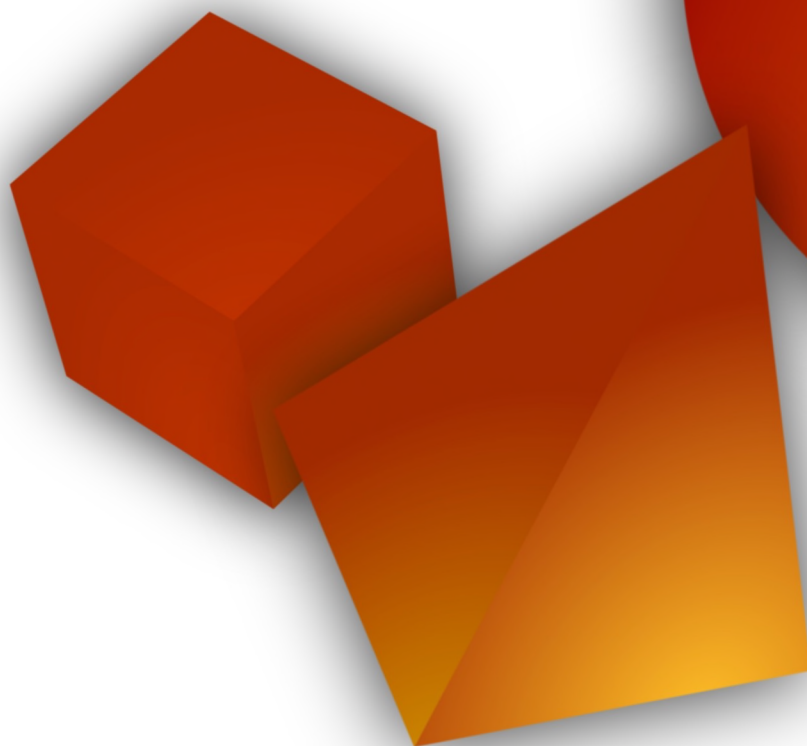


Benutzerhandbuch

VERSION

4.5



Borland[®] Turbo Profiler[®]

Borland C++

4.5

Turbo Profiler
Benutzerhandbuch

B O R L A N D

Copyright © 1987, 1994 by Borland International, Inc.
Alle Rechte vorbehalten.

Borland hat möglicherweise Patente oder Patentanmeldungen für Teile des hier behandelten Themas. Dieses Handbuch gibt Ihnen keine Rechte auf diese Patente.

Alle Borland-Produkte sind Warenzeichen oder eingetragene Warenzeichen von Borland International, Inc. Andere aufgeführte Waren- und Produktbezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Rechteinhaber.

Einschränkung der Gewährleistung:

Es wird keine Garantie für die Richtigkeit des Inhaltes dieses Handbuches übernommen. Da sich Fehler, trotz aller Bemühungen, nie vollständig vermeiden lassen, sind wir für Hinweise jederzeit dankbar.

Herausgeber:

Borland GmbH
Monzastraße 4c
63225 Langen

Satz, Layout & Acrobat-Bearbeitung:

ALFES Computers, Publishing & Graphics, Siegen

Printed in Ireland.

Inhalt

Einführung	11	Kapitel 2	
Der Unterschied zwischen Optimierung		Die Benutzeroberfläche	35
und Analyse	12	Teil 1: Die Bestandteile der	
Hardware- und Software-Anforderungen	13	Benutzeroberfläche	35
Neu in der Version 4.5	13	Die Menüleiste und die einzelnen Menüs	35
Der Inhalt dieses Handbuchs	13	Die Auswahl von Optionen mit der	
Typographische Konventionen	14	Tastatur	36
Software-Registrierung und technische		Die Auswahl einer Menüoption mit	
Unterstützung	15	der Maus	36
 Kapitel 1		Tastenkürzel	37
Ein Beispiel für eine Analyse	17	Fenster in Turbo Profiler	37
Beispielprogramme	18	Verwaltung der Fenster	38
Die Analyse eines Programms (PRIME0) ..	19	Die Statuszeile	39
Einstellung der Profiler-Optionen	20	Dialogfenster	39
Erfassung der Daten	20	Teil 2: Die Menüreferenz	40
Darstellung der Statistiken	21	System-Menü	40
Ausdrucken von Modulen und Statistiken ..	23	Repaint Desktop	40
Profiler Listing mit Angabe von Zeit und		Restore Standard	40
Anzahl	23	About	41
Reporte der Profiler-Statistik	25	File-Menü	41
Speichern und Laden von Statistiken	25	Open	41
Auswertung der Statistik	26	Protokollierungsdatei	42
Gleichzeitige Darstellung von Quellcode		Get Info	43
und Statistik	26	DOS Shell	44
Sichern der Fenster-Einstellung	27	Quit	44
Messen der Effizienz eines Bereichs ...	28	View-Menü	45
Ein modulares Primzahlen-Programm		Lokale Menüs	45
(PRIME1)	29	Module	46
Änderung des Programms und erneute		Line	47
Analyse	30	Search	47
Laden des nächsten Programms (PRI-		Next	48
ME2)	31	Goto	48
Verringerung der Aufrufe einer Funktion		Add Areas	48
(PRIME3)	31	Remove Areas	49
Weitere Verbesserungen in der Berech-		Operation	49
nung (PRIME4)	32	Callers	50
CR/LF eliminieren (PRIME6)	33	Module	51
Wie geht's weiter?	33	File	52
		Edit	52

Execution Profile	52	Text File	72
Display	53	Das lokale Menü des Fensters	
Filter	54	Text File	72
Module	55	Goto	72
Position	56	Search	72
Remove	56	Next	73
Callers	56	File	73
Inspect (linker Ausschnitt)	59	Edit	73
Inspect (rechter Ausschnitt)	59	Coverage-Analyse	73
Sort (rechter Ausschnitt)	59	Add All Modules (linker	
Overlays	60	Abschnitt)	74
Display	60	Remove All Modules (linker	
Inspect	61	Abschnitt)	74
Interrupts	61	Add Module (linker Abschnitt) .	74
Collection (oberer Ausschnitt) ..	62	Remove Module (linker	
Subroutine (oberer Ausschnitt) ..	62	Abschnitt)	75
Add (oberer Ausschnitt)	62	Delete This Item (linker	
Pick (oberer Ausschnitt)	62	Abschnitt)	75
Remove (oberer Ausschnitt)	62	Display (rechter Abschnitt)	75
Delete All (oberer Ausschnitt) ..	62	Position (rechter Abschnitt)	76
Display (unterer Ausschnitt)	62	Module (rechter Abschnitt)	76
Files	63	Run-Menü	77
Collection (oberer Ausschnitt) ..	63	Run	77
Detail (oberer Ausschnitt)	64	Program Reset	77
When Full (oberer Ausschnitt) ..	64	Arguments	77
Display (unterer Ausschnitt)	64	Das Menü Statistics	78
Areas	64	Callers	78
Add Areas	65	Files	79
Remove Areas	66	Interrupts	79
Inspect	66	Overlays	79
Options	66	Profiling-Optionen	80
Sort	67	Accumulation	81
Routines	67	Wann empfiehlt es sich, die	
Local Module (rechter Ausschnitt)	68	Erfassung abzuschalten?	82
Areas (beide Ausschnitte)	68	Delete All	84
Callers (beide Ausschnitte)	68	Save	84
Module (beide Ausschnitte)	68	Dateien speichern	85
Profile (beide Ausschnitte)	68	Restore	85
Disassembly (CPU)	69	Print-Menü	85
Goto	70	Statistics	86
Origin	70	Module	86
Follow	70	Options	87
Previous	71	Options-Menü	87
View Source	71	Makros	87
Mixed	71	Create	87

Stop recording	88	Was wollen Sie herausfinden?	103
Remove	88	Testen von Algorithmen	104
Delete All	88	Verifizierung und Testen	104
Makros aufzeichnen	88	Ausführungszeit und Ressourcen-Überwachung	105
Display Options	89	Untersuchung der Programmstruktur	105
Display Swapping	89	Welcher Analyse-Modus soll verwendet werden?	106
Screen Lines	89	Aktive Analyse	106
Tab Size	90	Passive Analyse	107
Width of Names	90	Was Sie immer im Auge behalten sollten	108
Path for Source	90	Coverage-Analyse	108
Save Options	90	Schnellere Analysen	108
Restore Options	91	Verbesserung der statistischen Genauig- keit	109
Window-Menü	92	Zuwenig Daten	109
Zoom	92	Resonanzen	109
Next	92	Einige Hinweise zur Analyse von Over- lays	110
Next Pane	92	Die Untersuchung objektorientierter Pro- gramme	111
Size/Move	92	Auswertung und Anwendung der Ergebnisse	111
Iconize/Restore	93	Wie werden die Profiler-Daten ausgewer- tet?	112
Close	93	Das Fenster Execution Profile	112
Undo Close	93	Das Fenster Callers	112
User Screen	93	Das Fenster Overlays	112
Liste der geöffneten Fenster	93	Das Fenster Interrupts	112
Das Menü Help	93	Das Fenster Files	112
Index	94	Das Fenster Coverage	113
Previous Topic	94	Filtern der erfaßten Daten	113
Help on Help	94	Überarbeiten Sie Ihr Programm	114
Kapitel 3		Ändern der Datenstrukturen	115
Analyse-Strategien	95	Speichern von vorausberechneten Ergebnissen	116
Vorbereitungen für die Analyse	96	Häufig verwendete Daten zwischenspeichern	116
Die Anpassung Ihres Programms	97	Nur erforderliche Daten auswerten ..	116
Compilierung des Programms	98	Optimierung bestehenden Codes ...	116
Bereiche festlegen	98	Zusammenfassung	118
Wie detailliert soll die Analyse sein?	100		
Weitere Bereiche in die Analyse aufnehmen	100		
Welche Daten wollen Sie untersuchen?	101		
Wann soll die Datenerfassung beginnen?	101		
Wie sollen die Zeitdaten zusammengefaßt werden?	102		
Welche Daten wollen Sie sehen?	102		
Die Analyse Ihres Programms	103		

Kapitel 4	
Die Funktionsweise von Turbo Profiler	119
Unsichtbare Mautstationen	120
Erfassen von Zeit- und Zählerwerten	120
Der Aufwand für einen	
Routinen-Aufruf	121
Was kostet eine Schleife?	121
Routinen mit mehreren Ausgängen	124
Die Erfassung von mehrfach aufgerufenen Routinen abschalten	125
Erfassung aufrufender Programmabschnitte	125
Stichproben oder Zählen	127
Die Speicherbelegung von Profiler	128
 Anhang A	
Die Kommandozeilen-Optionen	129
Kommandozeilen-Optionen	129
Stapelverarbeitungsbetrieb (-b)	130
Konfigurationsdatei (-c)	131
Darstellungsmodus (-d)	132
Hilfestellungen (-h und -?)	132
Protokoll speichern (-jn)	132
Mausunterstützung (-p)	132
Remote-Analysen (-r)	132
Quellcode und Symbole (-s)	133
Bildschirm-Optionen (-v)	134
 Anhang B	
Die Installation von Turbo Profiler	135
Der Aufruf von TFINST	136
Einstellung der Bildschirmfarben	136
Auswahl der Bildschirmfarben	136
Fenster	136
Dialogfenster und Menüs	137
Bildschirmelemente	137
Die vorgegebene Farbkombination	137
Die Bildschirmparameter von Turbo Profiler	137
Bildschirmumschaltung (Display Swapping)	138
Screen Lines	138
Fast Screen Update	138
Permit 51/58 Lines	138
Full Graphics Saving	139
Tab size	139
User Screen Updating	139
Die Optionen von Turbo Profiler	140
Das Dialogfenster Directories	140
Das Dialogfenster	
User Input & Prompting	140
History List Length	141
Interrupt Key	141
Mouse Enabled	141
Beep on Error	141
Control Key Shortcuts	141
Das Dialogfenster	
Miscellaneous Options	141
Printer Output	141
NMI-Intercept	142
Ignore Case of Symbol	142
International support	142
DOS Shell Swap Size (KByte)	142
Remote Type	143
Remote Link Port	143
Link Speed	143
Network local name	143
Network remote name	143
Einstellung des Bildschirmmodus	143
Default	143
Color	143
Black & White	143
Monochrome	144
LCD	144
Wenn Sie mit den Einstellungen fertig sind	144
Speichern der Änderungen	144
Save Configuration File	144
Modify TPROF.EXE	144
TFINST verlassen	145
Kommandozeilenoptionen und die entsprechenden TFINST-Optionen	145
 Anhang C	
Remote-Analysen	147
Voraussetzungen bei Hard-und Software	148
Remote-Analysen von DOS-Anwendungen	149
Das Remote-System einrichten	149
TFREMOTE einrichten	149
TFREMOTE anpassen	150

Der Remote-Treiber für DOS	150	Meldungen von TFREMOTE	158
Starten des seriellen Treibers auf dem		WREMOTE-Meldungen	161
Remote-System	151		
Starten des LAN-Treibers auf dem		Anhang D	
Remote-System	151	Turbo Profiler für Windows	163
DOS-Verbindung aktivieren	152	Die Installation von TPROFW	164
Die serielle Verbindung	152	Installation der Datei TDDEBUG.386 ..	164
LAN-Verbindungen	152	Das Konfigurieren von TPROFW	165
Remote-Analyse von		Der Kommandozeilenaufruf von TPROF	165
Windows-Anwendungen	153	Die Verwendung von TFINST im Zusam-	
Einrichten des Remote-Systems	153	menhang mit TPROFW	166
Die Konfiguration von WREMOTE	154	TPROFW im Einsatz	166
Serielle Konfiguration	154	Die Analyse von Windows-Prozeduren .	167
LAN-Konfiguration	154	Das Dialogfenster Window Procedure	
Die Kommandozeilenoptionen von		Messages	167
WREMOTE	155	Die Analyse von DLLs	169
Starten des Remote-Treibers für		Fehlermeldungen unter TPROFW	170
Windows	156		
Die Remote-Verbindung zum		Anhang E	
Windows-Rechner	156	Eingabeaufforderungen und	
LAN-Verbindung	156	Fehlermeldungen	173
Das Laden von Programmen auf das		Eingabeaufforderungen in Turbo Profiler ..	173
Remote-System	157	Fehlermeldungen von Turbo Profiler	177
Der Ablauf der Remote-Analyse	157		
Fehler und Probleme	158	Index	189

Abbildungen

Abbildung 1.1:		Abbildung 2.12:	
Turbo Profiler mit der Datei PRIME0	19	Das Dialogfenster <i>Area Options</i>	66
Abbildung 1.2:		Abbildung 2.13:	
Programm-Statistiken von PRIME0	20	Das Dialogfenster <i>Routines</i>	68
Abbildung 1.3:		Abbildung 2.14:	
Das Dialogfenster <i>Display Options</i>	21	Das Fenster <i>Disassembly (CPU)</i>	69
Abbildung 1.4:		Abbildung 2.15:	
Anzahl der Aufrufe im Fenster		Das Fenster <i>Coverage</i>	73
<i>Execution Profile</i>	22	Abbildung 2.16:	
		Das Dialogfenster <i>Coverage</i>	75
Abbildung 2.1:		Abbildung 2.17:	
Das Dialogfenster		Das Fenster <i>Coverage</i> ,	
<i>Load A New Program To Profile</i>	41	Gruppierung der Blöcke nach Routinen . . .	76
Abbildung 2.2:		Abbildung 2.18:	
Das Dialogfenster		Das Dialogfenster <i>Profiling Options</i>	80
<i>Enter Program Name To Load</i>	41	Abbildung 2.19:	
Abbildung 2.3:		Das Dialogfenster <i>Display Options</i>	89
Das Fenster <i>Module</i> (vergrößert)	46	Abbildung 2.20:	
Abbildung 2.4:		Das Dialogfenster <i>Save Configuration</i>	91
Das Dialogfenster <i>Area Options</i>	49		
Abbildung 2.5:		Abbildung B.1:	
Das Dialogfenster <i>Stack Trace</i>	50	Anpassung der Fensterfarben	136
Abbildung 2.6:		Abbildung B.2:	
Das Dialogfenster <i>Display Options</i>	53	Das Dialogfenster <i>Display Options</i>	137
Abbildung 2.7:		Abbildung B.3:	
Das Fenster <i>Callers</i> mit Aufrufen aus		Das Dialogfenster	
<i>CALLTEST</i>	56	<i>User Input and Prompting</i>	140
Abbildung 2.8:		Abbildung B.4:	
Das Fenster <i>Overlay</i>	60	Das Dialogfenster <i>Miscellaneous Options</i> .	141
Abbildung 2.9:			
Das Fenster <i>Interrupts</i>	61	Abbildung C.1:	
Abbildung 2.10:		WRSETUP-Hauptfenster und das	
Das Fenster <i>Files</i>	63	Dialogfenster <i>Settings</i>	154
Abbildung 2.11:			
Das Fenster <i>Areas</i>	65		

Tabellen

Tabelle 2.1: Tastenkürzel	37	Tabelle A.1: Die Kommandozeilenoptionen von Turbo Profiler	130
Tabelle 2.2: Fenster manipulieren	38	Tabelle B.1: Die Entsprechungen der Kommandozeilenoptionen	145
Tabelle 2.3: Speicheroptionen der Turbo Profiler- Protokollisten	43	Tabelle B.2: Optionen für TFINST.EXE	146
Tabelle 2.4: Übersicht über die Profiler-Fenster	45	Tabelle C.1: Die Kommandozeilenoptionen von TFREMOTE	149
Tabelle 2.4: Ausgabeformate der Interrupt-Statistiken ..	62	Tabelle C.2: Die Kommandozeilenoptionen von WREMOTE	155
Tabelle 3.1: Einsatzmöglichkeiten eines Profilers	104	Tabelle D.1: Die Kommandozeilenoptionen des Programms TPROF	165
Tabelle 3.2: Lokale Menübefehle zum Filtern von Statistiken	113	Tabelle D.2: Die Klassen der Windows-Meldungen	169

Einführung

Borlands Turbo Profiler ist das fehlende Glied in der Reihe Ihrer Software-Entwicklungswerkzeuge. Sobald Ihr Programm so funktioniert, wie Sie es möchten, können Sie Turbo Profiler einsetzen, um den Code schneller und effizienter zu gestalten.

Was ist unter einem Profiler zu verstehen? Profiler sind Software-Werkzeuge, mit denen man das Laufzeitverhalten eines Programms messen und anhand der Kern-Routinen herausfinden kann:

- wo ein Programm die meiste Zeit verbraucht;
- wie oft eine Zeile ausgeführt wird;
- welche Zeilen bereits ausgeführt wurden;
- wie oft und von wo aus eine Routine aufgerufen wird;
- auf welche Dateien ein Programm am häufigsten zugreift, und wieviel Zeit dafür aufgebracht wird.

Mit Turbo Profiler können Sie außerdem die wichtigsten Computer-Ressourcen überwachen:

- Prozessorzeit
- Dateizugriffe
- Tastatureingaben
- Druckausgaben
- Interrupt-Aktivitäten

Durch die Überwachung der wesentlichen Aktivitäten und die Durchführung detaillierter statistischer Auswertungen über jedes Detail des Laufzeitverhaltens Ihres Programms erhalten Sie die entscheidenden Hinweise zur Programmoptimierung. Turbo Profiler offenbart Ihnen die interne Funktionsweise Ihres Programms und schlüsselt die kompliziertesten Operationen auf – von der Anzahl einzelner Anweisungen sowie deren Ausführungszeit bis hin zu Interrupts und Dateizugriffen. Dadurch wird Turbo Profiler zu einer wesentlichen Stütze der Programmoptimierung.

Turbo Profiler zeichnet sich zwar durch eine Fülle an Funktionen aus, lässt sich aber trotzdem sehr einfach bedienen:

- Durch die interaktiv vorgenommene Analyse lassen sich ineffiziente Abschnitte sehr schnell erkennen.
- Beliebige Textdateien können während der Arbeit mit Turbo Profiler gelesen und editiert werden.
- Es können beliebig große Programme untersucht werden, die unter DOS oder Windows lauffähig sind.
- Programme, die mit Borland C++ Compilern oder Turbo Assembler erstellt wurden, lassen sich direkt verarbeiten.
- Die Benutzeroberfläche mit mehreren überlappenden Fenstern, Mausunterstützung und kontextabhängigen Hilfestellungen ist einfach zu bedienen.
- Häufigkeit und Dauer der Ausführung von Modulen, Funktionen und einzelnen Programmzeilen werden aufgezeichnet.
- Es läßt sich der gesamte Programmablauf mit den Aufrufen aller Funktionen erfassen sowie die Häufigkeit der Aufrufe mit vollständiger Zurückverfolgung der aufrufenden Programmabschnitte analysieren.
- DOS-Datei-Aktivitäten werden über ein Files-Fenster überwacht, in welchem die Kennung (Handle), der Zeitpunkt des Öffnens, Schließens, Lesens und Schreibens aller Dateien angezeigt wird. Im Protokoll wird neben dem Ablauf auch die Anzahl der gelesenen oder geschriebenen Bytes festgehalten.
- Alle Overlay-Zugriffe werden detailliert aufgezeichnet.
- Remote-Analysen lassen sich über die serielle Schnittstelle oder ein Netzwerk durchführen.

Turbo Profiler setzt dort an, wo Optimierungsprogramme ihre Arbeit einstellen. Sie werden sofort auf langsame Programmstellen und Engpässe hingewiesen und können diese gezielt durch bessere Algorithmen ersetzen.

Der Unterschied zwischen Optimierung und Analyse

Mit Hilfe eines Optimierungsprogramms können Sie ein Programm beschleunigen, indem Sie zeitaufwendige Befehle durch schnellere Befehle ersetzen. Nicht sauber geschriebene Programme lassen sich dadurch aber nicht verbessern.

Ein Profiler hingegen hilft Ihnen auf der Suche nach den Programmabschnitten, die am wenigsten effizient sind, sowie den Algorithmen, bei denen sich eine Überarbeitung lohnt. Untersuchungen haben ergeben, daß die größten Leistungsverbesserungen durch Änderungen von Algorithmen und Datenstrukturen erreicht werden, und nicht durch die Optimierung des Programmcodes. Die Suche nach den Programmengpässen ohne Profiler gleicht dem Versuch, einen Programmfehler ohne Debugger aufzuspüren. Turbo Profiler verringert nicht nur den Zeitaufwand, sondern stellt in manchen Fällen die einzige Chance dar, um eine bestimmte Optimierungsanforderung zu erfüllen.

**Hardware- und
Software-Anfor-
derungen**

Turbo Profiler stellt an die Hard- und Softwareumgebung die gleichen Anforderungen wie die von Ihnen verwendete Borland Programmiersprache. Eine komplette Auflistung dieser Anforderungen finden Sie im Benutzerhandbuch zu Ihrer Programmiersprache.

Neu in der Version 4.5

Turbo Profiler 4.5 verfügt gegenüber den vorhergehenden Versionen über verschiedene Erweiterungen: mehr Kapazitäten, Sicherung des Status einer Sitzung und DPMI-Analysen.

Turbo Profiler für Windows (TPROFW.EXE) kann größere Programme analysieren und stellt Statistiken für mehr Bereiche zur Verfügung, als dies mit früheren Versionen möglich war. Turbo Profiler kann ebenso den Status einer Sitzung speichern, mit dem Sie eine Analyse-Sitzung einfach wiederaufnehmen können. Zusätzlich verwendet TPROF.EXE (das Analyseprogramm unter DOS) nun das DOS Protected Mode Interface (DPMI), so daß Sie jetzt größere DOS-Programme analysieren können.

Der Inhalt dieses Handbuchs

Kapitel 1 Beispiel für eine Analyse ist ein Kurzlehrgang, in dem ein typischer (und doch einfacher) Ablauf einer Analyse behandelt wird. Dieses Kapitel fängt mit einer groben Analyse an, in der Sie einen Überblick gewinnen sollen. Anschließend werden die erfaßten Daten ausgewertet, um aufgrund der gewonnenen Einsichten das Programm zu ändern und zu verbessern. Danach werden weitere Analysen durchgeführt, um die Auswirkung weiterer Verbesserungen zu messen.

Kapitel 2 Die Benutzeroberfläche erklärt anhand von Analogien, wie der Profiler die Daten für die Ausführungszeit und die Anzahl der Ausführungen während des Programmablaufs erfaßt.

Kapitel 3 Analyse-Strategien liefert allgemeine Richtlinien zusammen mit einigen Hinweisen für sinnvolle Auswertungen.

Kapitel 4 Die Funktionsweise von Turbo Profiler erläutert die Bedeutung der einzelnen Menüpunkte und der Optionen von Dialogfenstern im Detail.

Anhang A Die Kommandozeilenoptionen enthält eine Beschreibung aller Kommandozeilenoptionen von Turbo Profiler.

Anhang B Die Installation von Turbo Profiler erklärt den Einsatz des Programms TFINST zum Einstellen und Ändern der Konfiguration von TPROF und TPROFW.

Anhang C Remote-Analysen beschreibt, wie man Analysen über zwei miteinander verbundene Systeme durchführen kann: auf dem einen System starten Sie das Programm, das untersucht werden soll, und auf dem anderen System untersuchen Sie das Programm mit Turbo Profiler.

Anhang D Turbo Profiler für Windows stellt dar, wie Turbo Profiler im Zusammenhang mit Windows eingesetzt wird und wie die speziellen Funktionen verwendet werden.

Anhang E Fehlermeldungen und Eingabeaufforderungen führt alle vorkommenden Eingabeaufforderungen und Fehlermeldungen auf und beschreibt, wie sie beantwortet werden können oder wie darauf reagiert werden kann.

Typographische Konventionen

In diesem Handbuch werden die folgenden speziellen Schriftarten verwendet:

Courier

Diese Schrift repräsentiert Text, den Sie eingeben oder der auf dem Bildschirm ausgegeben wird.

Kursiv

Diese Schrift dient zum Hervorheben einzelner Wörter, wie z.B. neuer Begriffe, sowie zur Kennzeichnung von Variablen, Funktionen, Klassen, Strukturen und Menüs.

Fett

Diese Schrift dient zur Kennzeichnung von reservierten Schlüsselwörtern, Formatbezeichnern und Kommandozeilenoptionen.

[Taste]

Dieser Typ bezeichnet eine Taste auf Ihrer Tastatur, die gedrückt werden soll. Zum Beispiel: „Drücken Sie [Entf], um das Zeichen zu löschen.“

[Taste1][Taste2]

Dieser Typ bezeichnet ein Tastenkürzel (engl. *hotkey*) eines Befehls. Dies bedeutet, daß Sie [Taste1] zusammen mit [Taste2] drücken sollen. Zum Beispiel: „Drücken Sie [Alt][X], um das Programm zu beenden.“

VERSALIEN

Dieser Typ repräsentiert Laufwerksverzeichnisse, Dateinamen und Anwendungsbezeichnungen (Namen von Header-Dateien werden jedoch in kleinen Buchstaben dargestellt, um die Konsistenz mit den Quelltexten sicherzustellen).

Menü / Auswahl

Dieser Typ bezeichnet Menübefehle. In den Borland Handbüchern wird die Konvention „Wählen Sie *Datei* / *Speichern*“ anstelle von „Wählen Sie den Befehl *Speichern* aus dem Menü *Datei*“ verwendet.

Software-Registrierung und technische Unterstützung

Das Borland Assist Programm bietet eine Reihe von technischen Unterstützungen an, die auf die Bedürfnisse von Entwicklern, großen Firmen, Beratern und Programmierern zugeschnitten sind. Senden Sie uns Ihre Registrierungskarte, und wählen Sie den Borland Assist Plan aus, der am besten auf Sie zugeschnitten ist. Weitere Informationen über diese und andere Dienstleistungen von Borland finden Sie im Borland Übersichtsfaltblatt, das diesem Produkt beigelegt ist.

Ein Beispiel für eine Analyse

Die Analyse des Laufzeitverhaltens eines Programms ist einer der am wenigsten verstandenen, gleichwohl aber einer der wichtigsten Bereiche in der Software-Entwicklung. Umfragen deuten darauf hin, daß nur wenige professionelle Programmierer tatsächlich einen Profiler einsetzen, um den Code zu verbessern. Des weiteren zeigte sich, daß auch die besten Programmierer meist eine falsche Vorstellungen über die Engpässe in ihren Programmen haben.

Wo liegen also die Vorteile dieses weitgehend unbeachteten Werkzeugs? Zum einen kann man durch eine Laufzeitanalyse die Gesamtleistung eines Programms verbessern. Zum anderen verbessert sich durch die stetige Analyse von Programmen nach und nach die Fähigkeit, guten Programmcode zu erzeugen. Als Schlußfolgerung bietet sich der konsequente Einsatz eines Profilers innerhalb der Programmentwicklung an.

Die Beispiele in diesem Kapitel wurden Jon Bentleys Kolumne „Programming Pearls“ (Juli 1987) aus *Communications of the ACM* entnommen.

In diesem Kapitel zeigen wir anhand eines Beispiels, wie man einen Profiler vorteilhaft einsetzen kann und wie man – langfristig – bei der Suche nach ineffizientem Code viel Zeit sparen kann. Mit Turbo Profiler können Sie:

- feststellen, wo Ihr Programm die Zeit verbraucht;
- ein kommentiertes Quellcode-Listing und eine Profiler-Statistik ausgeben;
- Profiler-Statistiken speichern und eine unterbrochene Sitzung mit bereits gewonnenen Erkenntnissen fortsetzen;
- Profiler-Statistiken dem Quellcode in verschiedenen Fenstern gegenüberstellen und auswerten.

Alle Beispiele wurden auf einem 80486-System mit SVGA-Adapter getestet.

In den Beispielen dieses Kapitels geht es um die Suche und Ausgabe aller Primzahlen zwischen 1 und 1000. Eine Primzahl ist nur durch sich selbst und die Zahl 1 teilbar. Sie ist mit Ausnahme der Zahl 2 immer ungeradzahlig, da jede gerade Zahl auf alle Fälle durch 2 teilbar ist.

Das Ziel der Analyse dieses Beispielprogramms ist die Beschleunigung der Ermittlung und der Ausgabe dieser Primzahlen. Um herauszufinden, ob eine Zahl eine Primzahl ist, prüft man, ob sie durch andere, kleinere Primzahlen oder eine Zahl teilbar ist, die größer als die ersten beiden Primzahlen, 2 und 3, ist. Wenn Sie diese Dateien durcharbeiten, werden Sie lernen, wie man mit Hilfe des Profilers die Effizienz der in den Beispielen verwendeten Strukturen testen kann.

Das erste Programm, das wir analysieren, heißt PRIME0. Wenn Sie dieses Programm untersucht haben und wissen, wie und an welchen Stellen der Code geändert werden sollte, brauchen Sie bloß PRIME1 zu laden. Mit Ausnahme von PRIME1 ist jedes der Programme (PRIME2, PRIME3, PRIME4 und PRIME5), die in diesem Kapitel behandelt werden, eine Abwandlung des vorhergehenden Programms.

Beispielprogramme

Alle Beispiele liegen im Quellcode und als ausführbares Programm vor. Turbo Profiler benötigt beide Dateien, um ein Programm untersuchen zu können. Jedes Beispiel wurde so kompiliert, daß alle symbolischen Informationen hinzugefügt wurden, da auch der Profiler diese Informationen benötigt.

Damit Ihre Programme alle symbolischen Informationen enthalten, compilieren Sie die Dateien mit folgenden Compiler-Optionen:

- Borland C++ Compiler: Wenn Sie in der integrierten Entwicklungsumgebung compilieren,
 1. Wählen Sie *Optionen / Projekt*.
 2. Wählen Sie *Compiler / Debugger*, markieren Sie *Debug Informationen in OBJ-Dateien*.
 3. Wählen Sie *Linker / Allgemein*, markieren Sie *Debug Informationen mit aufnehmen*.

Wenn Sie von der Kommandozeile aus compilieren, verwenden Sie die Kommandozeilenoption `-v`.

- Turbo Assembler: Sie geben die Kommandozeilenoption `/zi` an und linken das Programm unter Angabe der Option `/v` mit TLINK.

Die Analyse eines Programms (PRIME0)

Die Analyse und Verbesserung eines Programms umfaßt im Grunde genommen vier Schritte:

1. Programm für die Analyse vorbereiten.
2. Daten erfassen während des Programmlaufs.
3. Erfasste Daten auswerten.
4. Programm ändern, anschließend neu compilieren.

Nachdem Sie Ihr Programm geändert haben, wiederholen Sie die Schritte 1-3, um festzustellen, ob sich die Laufzeiteigenschaften Ihres Programms weiter verbessern lassen.

PRIME0 verwendet die Euklid'sche Methode für die Prüfung auf Primzahlen: eine einfache Integer-Prüfung auf den Rest nach einer Division. Nachdem eine Primzahl ermittelt wurde, wird sie im Array Primzahlen gespeichert. Alle nachfolgenden Zahlen werden beim Primzahltest durch alle Zahlen geteilt, die bereits in Primzahlen gespeichert sind.

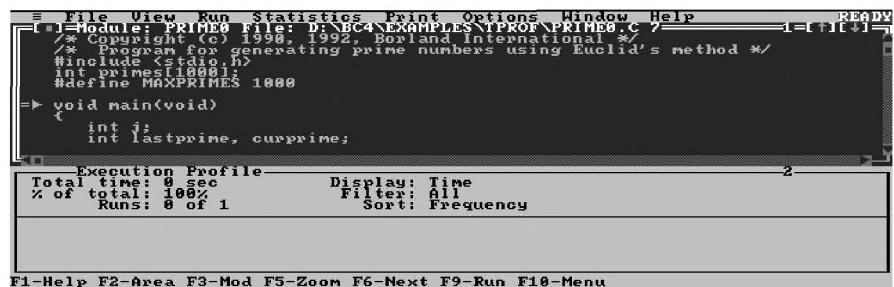
Mit *File/Quit* oder der Tastenkombination [ALT][X] können Sie Turbo Profiler jederzeit wieder verlassen.)

Nach der Eingabe des Befehls

```
TPROF PRIME0
```

meldet sich der Profiler mit zwei geöffneten Fenstern: das *Module*-Fenster (in dem der Quellcode von PRIME0 zu sehen ist) und das Fenster *Execution Profile* (das nach der Ausführung von PRIME0 die Statistiken enthält).

Abbildung 1.1:
Turbo Profiler mit
der Datei PRIME0



Eine detaillierte Beschreibung der Benutzeroberfläche von Turbo Profiler finden Sie in Kapitel 2.

Die Fenster *Module* und *Execution Profile* entsprechen den Ergebnissen der Schritte 1 und 3 des Ablaufs einer Analyse. Sie verwenden das Fenster *Module*, um festzulegen, welcher Programmabschnitt untersucht werden

soll. Sobald das Programm läuft, werden im Fenster *Execution Profile* die gewünschten Informationen dargestellt, mit denen Sie das Programmverhalten beurteilen können.

Einstellung der Profiler-Optionen

Bevor Sie mit der Analyse Ihres Programms beginnen, müssen Sie die Bereiche angeben, die untersucht werden sollen. Ein *Bereich* ist ein Abschnitt in Ihrem Programm, über den Sie Statistiken erfassen wollen. Ein Bereich kann eine einzelne Zeile, ein Programmkonstrukt wie eine Schleife oder eine komplette Routine sein.

Falls Sie wenige kurze Funktionen (wie die Funktionen **prim** oder **main**) dieses Programms untersuchen wollen, möchten Sie vielleicht wissen, wie oft jede Zeile ausgeführt wird und wieviel Zeit dafür benötigt wird. Um diese Informationen zu erhalten, müssen Sie jede Programmzeile als eigenen Bereich markieren:

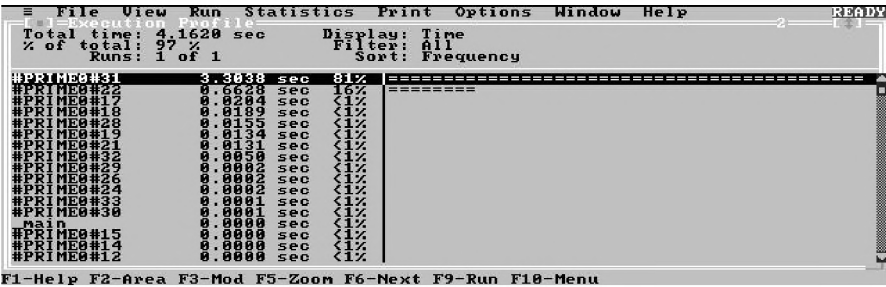
1. Drücken Sie *[Alt][F10]*, um das lokale Menü des Fensters *Module* zu öffnen.
2. Wählen Sie den Befehl *Add Areas* (Bereiche aufnehmen) im lokalen Menü aus. Die nachfolgend erscheinende Liste enthält die Bereichsgrenzen, die Sie auswählen können.
3. Wählen Sie dann im Fenster *Module* die Option *Every Line* (jede Zeile). Diese Aktion setzt die Bereichsmarkierungen auf alle Zeilen im Modul und aktiviert dann wieder das Fenster *Module*.

Erfassung der Daten

Nun sind wir ausreichend auf den zweiten Schritt der Analyse vorbereitet. Drücken Sie *[F9]*, um PRIME0 unter Turbo Profiler zu starten. Wie Sie sehen, gibt das Programm die Primzahlen zwischen 1 und 1000 auf dem Bildschirm aus. Nach Abschluß des Programms sehen Sie die Profiler-Statistiken im Fenster *Execution Profile*. Dies sind Ihre *Programmstatistiken*.

Vergrößern Sie das Fenster *Execution Profile*: Drücken Sie *[F5]* oder wählen Sie den Befehl *Zoom* im Menü *Window*. Ihr Bildschirm sollte nun folgendermaßen aussehen:

Abbildung 1.2:
Programm-Statistiken von PRIME0



Der obere Ausschnitt des Fensters *Execution Profile* stellt die gesamte Ausführungsdauer des Programms zusammen mit den Informationen über die Daten des unteren Ausschnitts dar. Der untere Ausschnitt hat in jeder Zeile vier Einträge:

- einen Bereichsnamen;
- die Anzahl an Sekunden, die in diesem Bereich zugebracht wurden;
- den Prozentsatz der gesamten Ausführungszeit in diesem Bereich;
- einen Balken, der die Größenordnung des Prozentsatzes wiedergibt.

Die Zeile

```
#PRIME0#31    7.2493 sec  85%  |=====
```

sagt aus, daß die 31. Programmzeile im Modul PRIME0 ungefähr 7,2 Sekunden lang ausgeführt wurde – das entspricht etwa 85 Prozent der gesamten Ausführungsdauer aller markierten Bereiche. Der Balken für die Größenordnung wird automatisch voll ausgedehnt (da PRIME0#31 der zeitintensivste aller markierten Bereiche ist).

Die tatsächlichen Zeit- und Prozentwerte können von System zu System unterschiedlich sein.

Darstellung der Statistiken

Sie können alle Meßdaten auch in Form der *Ausführungsanzahl* der einzelnen Bereiche darstellen.

1. Drücken Sie **[Alt][F10]**, um das lokale Menü des Fensters *Execution Profile* zu öffnen.
2. Wählen Sie den Befehl *Display* im lokalen Menü.
3. Das Dialogfenster mit den Darstellungsoptionen (*Display Options*) stellt fünf verschiedene Methoden für die Anzeige der Daten im Fenster *Execution Profile* zur Auswahl.

Abbildung 1.3:
Das Dialogfenster
Display Options



- *Time* (Zeit) zeigt die Ausführungszeit der einzelnen markierten Bereiche an. (Dies ist auch die Standard-Einstellung.)
- *Counts* (Anzahl) gibt an, wie oft das Programm jeden Bereich ausgeführt hat.
- *Both* (beides) stellt sowohl die Anzahl als auch die Zeitdaten im selben Bildschirmbereich dar.

- *Per Call* (pro Aufruf) gibt den durchschnittlichen Zeitbedarf pro Aufruf wieder.
 - *Longest* (am längsten) zeigt die längste Zeit an, die jeweils in den einzelnen Bereichen zugebracht wurde.
 - *Modules* (Module) wird bei der passiven Analyse verwendet und gibt den Zeitbedarf an, der für jedes einzelne Modul anfällt.
4. Wählen Sie in diesem Dialogfenster die Option *Counts*. (Klicken Sie den Menüpunkt *Counts* mit der Maus an oder bewegen Sie den Auswahlbalken mit den Pfeiltasten dorthin und drücken Sie *[Eingabe]*, oder drücken Sie einfach nur die Taste *[C]*, die Abkürzungstaste für dieses Menü.)
 5. Wählen Sie *OK* (oder drücken Sie *[Eingabe]*).

Das Fenster *Execution Profile* stellt nun die Statistik für PRIME0 als Anzahl von Aufrufen anstelle der Ausführungszeiten dar, wie Sie folgender Abbildung entnehmen können:

Abbildung 1.4
Anzahl der Aufrufe
im Fenster
Execution Profile

Module	Count	Percentage
PRIME0#22	15122	82%
PRIME0#19	500	2%
PRIME0#28	499	2%
PRIME0#21	499	2%
PRIME0#26	333	1%
PRIME0#23	333	1%
PRIME0#24	333	1%
PRIME0#31	166	<1%
PRIME0#32	166	<1%
PRIME0#33	166	<1%
PRIME0#30	166	<1%
Main	1	<1%
PRIME0#18	1	<1%
PRIME0#17	1	<1%
PRIME0#15	1	<1%
PRIME0#14	1	<1%
PRIME0#12	1	<1%

Diese Darstellung der Statistik von PRIME0 zeigt, daß Zeile 22 in PRIME0 am häufigsten aufgerufen wurde.

Sie können sich die Statistiken über Anzahl und Zeit auch zusammen ansehen. Sie können das machen, indem Sie dasselbe Dialogfenster noch einmal aufrufen. (Drücken Sie entweder *[Alt][F10]* und wählen Sie den Menüpunkt *Display*, oder drücken Sie *[Strg][D]*.)

Wählen Sie in *Display* die Option *Both* (Beides) und dann *OK* oder drücken Sie *[Eingabe]*. (Für *Both* klicken Sie den entsprechenden Menüpunkt entweder mit der Maus an oder bewegen Sie den Auswahlbalken mit *[Nach unten]* dorthin und drücken *[Eingabe]*. Sie können aber auch die Abkürzungstaste *[B]* verwenden.)

Wenn im Fenster *Execution Profile* Zeit und Anzahl zusammen dargestellt werden, enthält der erste Eintrag eines jeden Bereiches die Anzahl der Aufrufe und der zweite die Zeit.

Ausdrucken von Modulen und Statistiken

In diesem Abschnitt lernen Sie, zwei Arten von Informationen auszugeben:

1. ein Profiler-Listing des Quellcodes, der sich im Fenster *Module* befindet, einschließlich der Angaben über Zeiten und Anzahl der Durchläufe in den markierten Bereichen.
2. die Profiler-Statistik im Fenster *Execution Profile*.

Profiler Listing mit Angabe von Zeit und Anzahl

Bevor Sie ein Profiler Listing mit den Werten von Zeit und Anzahl in eine Datei schreiben, müssen Sie die Druck-Optionen korrekt einstellen:

1. Wählen Sie den Befehl *Print / Options*.
2. Im Dialogfenster *Printing Options* (Druck-Optionen) wählen Sie die Option *File* (Datei). Drücken Sie die Taste *[Tab]*, bis die gewünschten Optionen aktiviert sind und drücken Sie anschließend *[Nach unten]*, um die Einstellung *File* vorzunehmen.
3. Wechseln Sie mit *[Tab]* zum Dialogfenster für die Ausgabedatei (*Destination File*) und geben Sie folgenden Dateinamen ein:

```
PRIME0SC.LST
```

4. Wählen Sie die Option *ASCII*, um den Standard-ASCII-Zeichensatz zu verwenden (und nicht den erweiterten IBM-Zeichensatz).
5. Bestätigen Sie die Auswahl mit OK (oder drücken Sie *[Eingabe]*).

Der Cursor kehrt dann zum aktiven Fenster *Execution Profile* zurück.

Nun wählen Sie den Befehl *Print / Module* (Drucken/Modul), um den Ausdruck vorzunehmen. Im Dialogfenster mit dem Titel *Pick a Module* (Auswahl eines Moduls) drücken Sie *[Nach unten]*, um die Modulbezeichnung *PRIME0* mit dem Auswahlbalken hervorzuheben. Drücken Sie dann *[Eingabe]* (oder wählen Sie OK).

Um sich die Datei *PRIME0SC.LST* anzusehen, wählen Sie *View / Text File*; auf die entsprechende Aufforderung hin geben Sie den Dateinamen ein:

```
PRIME0SC.LST
```

Sie erhalten dann folgende Statistik für das untersuchte C-Programm *PRIME0*. Beachten Sie bitte, daß sich die Zeitangaben in Ihrer Datei wahrscheinlich von den hier angegebenen Zeiten aufgrund unterschiedlicher Computer-Konfigurationen unterscheiden:

Turbo Profiler Version4.5 Tue Aug 20 15:16:47 1994

Program: D:\TPROF\PRIME0.EXE File PRIME0.C

```

Time    Counts
        /* Copyright (c) 1990, Borland International */
        /* Programm zur Ermittlung von Primzahlen nach der */
        /* Methode von Euklid */

        int Primzahlen[MAXPRIMZAHL];
        #define MAXPRIMZAHL 1000

0.0000 1  main()
        {
            int j;
            int letzte_Prim, akt_Prim;

0.0000 1      Primzahlen[0] = 2;
0.0000 1      Primzahlen[1] = 3;
0.0000 1      letzte_Prim = 1;
0.0000 1      akt_Prim = 3;

0.0359 1      printf("prime %d = %d\n", 0, Primzahlen[0]);
0.0354 1      printf("prime %d = %d\n", 1, Primzahlen[1]);
0.0059 500    while(akt_Prim < MAXPRIMZAHL)
        {
0.0071 499        for(j = 0; j <= letzte_Prim; j++)
0.3069 15122        if((akt_Prim % Primzahlen[j]) == 0)
            {
0.0038 333            akt_Prim += 2;

0.0034 333            break;
            }
0.0060 499        if(j <= letzte_Prim)
0.0037 333            continue;
0.0077 166        letzte_Prim++;
6.2655 166        printf("Primzahl %d = %d\n", letzte_Prim, akt_Prim);
0.0019 166        Primzahlen[letzte_Prim] = akt_Prim;
0.0018 166        akt_Prim += 2;
            }
0.0000 1      }

```

Diese mit dem Profiler erstellte Datei ist sehr nützlich, da es sich um eine permanente Aufzeichnung handelt, die für jeden Bereich Ihres Programms die Ausführungszeiten und die Anzahl der Durchläufe speichert.

Nachdem Sie sich diese Liste genau angesehen haben, drücken Sie **[Alt][F3]** oder klicken auf *Schließen*, um das File-Fenster zu schließen.

**Reporte der
Profiler-Statistik**

Sie können eine Kopie des Fensterinhalts von *Execution Profile* auch direkt über den Drucker oder in eine Datei ausgeben:

1. Wählen Sie nochmals den Befehl *Print / Options*.
2. Wählen Sie die Option *Printer* (Drucker).
3. Wählen Sie *Graphics* (Grafik), damit die IBM-Grafikzeichen im Bericht verwendet werden. (Wenn Ihr Drucker keine IBM-Grafikzeichen unterstützt, lassen Sie diesen Schritt aus und machen mit Punkt 4 weiter).
4. Geben Sie [*Eingabe*] ein (oder wählen Sie *OK*).
5. Wählen Sie anschließend den Befehl *Print / Statistics* (Drucken/Statistiken).

Der Ausdruck ist wie das Listing der Quelldatei eine dauerhafte Aufzeichnung, die für die weitere Suche nach dem kompaktesten und effizientesten Code für Ihr Programm nützlich ist.

Speichern und Laden von Statistiken

Bevor Sie weitermachen, zeigen wir Ihnen, wie man die Profiler-Statistik von PRIME0 in einer Datei speichern kann, damit Sie eine Sitzung mit Turbo Profiler jederzeit unterbrechen können, ohne damit die gewonnenen Daten zu verlieren. Anschließend zeigen wir Ihnen, wie Sie diese Statistik beim nächsten Aufruf von Turbo Profiler wieder laden können.

Wählen Sie den Befehl *Statistics / Save* (Statistiken/Speichern), um die Profiler-Statistik Ihres Programms in einer .TFS-Datei (Turbo Profiler Statistik-Datei) zu speichern. Da sich PRIME0 im *Module*-Fenster befindet, erscheint der Dateiname PRIME0.TFS im Eingabefeld *File Name* (Dateiname) automatisch als Vorgabe. Wählen Sie *OK*, um diese Datei zu erstellen. Alle statistischen Daten der aktuellen Analyse von PRIME0 werden daraufhin in der Datei PRIME0.TFS im aktuellen Verzeichnis gespeichert.

Um die Statistikdaten von PRIME0 erneut in den Profiler zu laden, wählen Sie, nachdem Sie PRIME0 geladen haben, den Befehl *Statistics / Restore* (Statistiken/Laden).

Im Eingabefeld *File Name* steht PRIME0.TFS wieder als Voreinstellung. Drücken Sie [*Eingabe*] oder wählen Sie *OK*, um die Daten aus der Datei zu übernehmen.

Auswertung der Statistik

In diesem Abschnitt lernen Sie, die Statistiken im Fenster *Execution Profile* auszuwerten, um damit Ihr Programm verbessern zu können.

Zunächst verkleinern Sie das Fenster *Execution Profile* (wählen Sie *Zoom* im *Window*-Menü, oder drücken Sie [F5]). Anschließend betrachten Sie die Statistik für die Zeilen 22 und 31 (die **if**- und **printf**-Anweisungen).

Wir behandeln die Änderungen für die **printf**-Anweisung im Rahmen des Programms PRIME5.

Eine Zeit- / Anzahl-Analyse wie diese sagt eine ganze Menge über das Programm aus. Sie können zum Beispiel erkennen, daß Zeile 22 in PRIME0 viel öfter als jede andere Anweisung ausgeführt wird. Es scheint richtig zu sein, daß Zeile 22 15122-mal ausgeführt wird, da jede Zahl zwischen 4 und 1000 mit allen anderen Zahlen im Array **Primzahlen** verglichen wird, bis eine Division aufgeht oder das Fassungsvermögen des Arrays erschöpft ist. Das bedeutet, daß sehr viele Zahlen miteinander verglichen werden müssen. Sie erkennen außerdem, daß Zeile 31, die **printf**-Anweisung, den größten Teil der gesamten Ausführungszeit des Programms in Anspruch nimmt.

Gleichzeitige Darstellung von Quellcode und Statistik

Die Daten im Fenster *Execution Profile* zeigen, daß die Abfrage in Zeile 22 mehr macht, als eigentlich nötig ist. Aber Sie erhalten keinen umfassenden Überblick, wenn Sie die Daten für Anzahl und Zeit nicht zusammen betrachten.

Sie müssen also die Daten im Fenster *Execution Profile* und den entsprechenden Abschnitt des Quellcodes im *Module*-Fenster vergleichen.

Um Quellcode und Profiler-Statistik gleichzeitig betrachten zu können, gehen Sie wie folgt vor:

1. Verändern und verschieben Sie das Fenster *Execution Profile* so, daß es die rechte Hälfte Ihres Bildschirms belegt: Wählen Sie *Window / Size/Move* oder drücken Sie [Strg][F5].
2. Folgen Sie den Anweisungen in der Statuszeile, um:
 - a) das Fenster auf volle Höhe und halbe Breite des Bildschirms zu vergrößern;
 - b) das veränderte Fenster auf die rechte Seite zu verschieben.

Wenn Sie mit Schritt a und b fertig sind, drücken Sie [Eingabe].

3. Aktivieren Sie das *Module*-Fenster mit [F6], vergrößern Sie es anschließend und verschieben Sie es so, daß es die linke Hälfte des Bildschirms ausfüllt.
4. Wechseln Sie wieder ins Fenster *Execution Profile* (indem Sie erneut [F6] drücken).

Um die Größe eines Fensters mit der Maus zu verändern, verschieben Sie das Größenänderungsfeld in der rechten unteren Ecke des Fensters. Zum Verschieben des Fensters klicken Sie das Fenster an der Titelleiste oder der linken bzw. der oberen Doppellinie an und ziehen Sie es auf die gewünschte Position.

Nun können Sie die Verbindung vom Fenster *Execution Profile* zum *Module*-Fenster herstellen, den Quellcode durchgehen und die Analyse-Daten mit den entsprechenden Programmzeilen im Quellcode vergleichen:

1. Aktivieren Sie das Fenster *Execution Profile* (mit [F6]) und verschieben Sie den Auswahlbalken auf die erste Zeile des Fensters.
2. Öffnen Sie das lokale Menü (drücken Sie [Alt][F10]) und wählen Sie *Module* (oder drücken Sie das Tastenkürzel [Strg][M]).

Der Profiler positioniert den Cursor auf die Zeile 31 im Fenster *Module*.

3. Verwenden Sie die Pfeiltasten, um im Quellcode zur Zeile 22 zu gelangen (Zeile 31 in PRIME0PA).

Diese Zeile ist der Bereich mit dem zweitgrößten Zeitbedarf in PRIME0. Die beiden oberen Zeilen der Statistik im Fenster *Execution Profile* enthalten nun die Profiler-Daten für diese **if**-Anweisung.

4. Verschieben Sie den Cursor im Fenster *Module* auf Zeile 21 (Zeile 29 in PRIME0PA) und beachten Sie dabei, wie die Darstellung im Fenster *Execution Profile* automatisch nachgezogen wird: In den oberen Zeilen des Fensters *Execution Profile* steht nun die Profiler-Statistik für Zeile 21.
5. Bewegen Sie den Cursor in die Zeile 30 (Zeile 42 in PRIME0PA) und beobachten Sie gleichzeitig die Darstellung im Fenster *Execution Profile*.

Durch eine derartige Synchronisation der beiden Fenster ist es leichter, den größten Verbrauch von Ressourcen in Ihrem Programm herauszufinden. Sobald Sie sich an die Auswertung der Daten am Bildschirm gewöhnt haben, brauchen Sie nicht mehr so oft auf Profiler-Ausdrucke zurückgreifen.

Sichern der Fenster-Einstellung

Der Zeitpunkt ist günstig, diese spezielle Einstellung von Turbo Profiler abzuspeichern. (Wenn Sie Ihre speziellen Fenster-Anordnungen nicht speichern, erhalten die Fenster wieder Ihre ursprüngliche Größe und Anordnung, wenn Sie das nächste Programm in Turbo Profiler laden.)

1. Wählen Sie den Befehl *Options / Save Option* (Optionen/Optionen speichern). Damit öffnen Sie das Dialogfenster *Save Configuration* (Einstellung speichern).
2. Das Markierungsfeld *Options* ist per Voreinstellung bereits markiert. Damit werden die Einstellungen (wie die Darstellungsoptionen des Fensters *Execution Profile*) für die nächste Speicherung in die Konfigurationsdatei vorgemerkt.
3. Im Dialogfenster *Save Configurations* wählen Sie *Layout*. Damit wird der Aufbau der nebeneinanderliegenden Fenster für die nächste Speicherung in die Konfigurationsdatei vorgemerkt.
4. Normalerweise wird für die Konfigurationsdatei der Name *TFCONFIG.TF* benutzt, der im Eingabefeld *Save To* (Speichern in) automatisch vorgegeben wird. Wählen Sie *OK* oder drücken Sie [*Eingabe*], um Ihre Optionen im aktuellen Verzeichnis zu speichern.

Beim nächsten Start von Turbo Profiler wird die Datei *TFCONFIG.TF*, die Standard-Konfigurationsdatei, gesucht und – sofern vorhanden – eingelesen. Dadurch werden die Anordnung der Fenster und die dort eingestellten Optionen automatisch übernommen.

Messen der Effizienz eines Bereichs

Das Verhältnis der Ausführungszeit zur Anzahl der Durchläufe eines Bereiches ist ein gutes Maß für die Effizienz einer Zeile oder der Gesamteffektivität einer Routine. Um dieses Verhältnis für die Bereiche in *PRIME0* zu ermitteln, brauchen Sie bloß die Darstellungsoption im Fenster *Execution Profile* zu verändern. Das geht folgendermaßen:

1. Im lokalen Menü des Fensters *Execution Profile* ([*Alt*][*F10*]) wählen Sie *Display*.
2. Im Dialogfenster von *Display* wählen Sie die Option *Per Call* (pro Aufruf).
3. Bestätigen Sie mit *OK* (oder drücken Sie [*Eingabe*]).

Nun sehen Sie, daß ungeachtet der Tatsache, daß Zeile 22 viel öfter als Zeile 31 aufgerufen wird (Zeile 31 und 42 in *PRIME0PA*), Zeile 22 viel effektiver als Zeile 31 ist. Die Zeile beansprucht zwar viel Zeit, weil sie so oft aufgerufen wird, aber jeder einzelne Aufruf dauert im Durchschnitt nicht länger als eine Millisekunde. Dagegen benötigt Zeile 31 im Schnitt fast 38 Millisekunden pro Aufruf (in *PRIME0PA* benötigt Zeile 42 durchschnittlich 28 Millisekunden).

Hinweis

Die Ausgabe des Profilers weist uns den Weg, wie wir die Ausführungszeit von *PRIME0* verbessern und das Programm strukturell vereinfachen können. Die Verbesserung des Programms kann auf zwei Arten erfolgen:

1. Verringerung des Zeitbedarfs für die Ein-/Ausgabe.
2. Schreiben einer effizienteren Schleife.

Das Ein-/Ausgabe-Problem läßt sich zum Teil dadurch lösen, daß man die **printf**-Anweisung von der jetzigen Form

```
printf ("Primzahl #%d = %d\n", letzte_Prim, akt_Prim);
```

einfach wie folgt verkürzt:

```
printf ("%d\n", akt_Prim);
```

Diese einfache Änderung bedeutet bereits eine enorme Verringerung der Ausführungsdauer. Die Anzahl der Aufrufe der Ausgabeanweisung können Sie damit aber nicht ändern. In der gegebenen Konstellation müssen immer 168 Primzahlen ausgedruckt werden. Und außer dieser einfachen Verbesserung gibt es nicht mehr allzu viele Möglichkeiten, um die Ausführung von PRIME0 zu beschleunigen. Der Algorithmus, durch den alle Ergebnisse in einem Array gespeichert und anschließend für die Division verwendet werden, ist gründlich, aber kaum zu verbessern. (Er ist zudem nicht sehr speichereffizient, da das Array so groß dimensioniert werden muß, um alle Primzahlen aufzunehmen, die ermittelt werden sollen. Dadurch wird eventuell die Anzahl an Primzahlen eingeschränkt, die getestet werden können, ohne daß der Speicher überläuft.)

Es gibt jedoch zum Glück noch eine bessere Methode zur Ermittlung von Primzahlen: mit einem anderen Algorithmus. Dieser kommt im nächsten Beispielprogramm, PRIME1, zum Einsatz.

Ein modulares Primzahlen-Programm (PRIME1)

Sie sind nun mit PRIME0 fertig. Laden Sie also das nächste Beispielprogramm, PRIME1, die nächste Version des Primzahlen-Programms, in das *Module*-Fenster und schauen Sie sich den Code an:

1. Wählen Sie *File / Open*.
2. Standardmäßig ist das Eingabefeld *File Name* aktiviert und enthält die Dateimaske *.EXE. Drücken Sie [*Eingabe*].
3. Verwenden Sie die Pfeiltasten, [*Nach unten*] und [*Nach oben*], um die Datei PRIME1.EXE aus der Dateiliste auszuwählen.
4. Drücken Sie [*Eingabe*]. Turbo Profiler lädt PRIME1 (PRIME1PA) in das *Module*-Fenster.
5. Vergrößern Sie das *Module*-Fenster (mit [*F5*]). Beachten Sie die hinzugekommene Funktion Prim in Zeile 4.

Wie Sie sehen, wurden im wesentlichen zwei Änderungen durchgeführt:

- Das Array **Primzahlen** ist nicht mehr vorhanden. Dieses Programm ermittelt die Primzahlen nicht über eine Division durch die kleineren Primzahlen, vielmehr verwendet es einfach eine Schleife, um die untersuchte Zahl durch alle kleineren ungeraden Zahlen zu teilen. Dieser Algorithmus erfordert anfangs etwas mehr Iterationen, aber wie Sie sehen werden, ergibt sich dadurch ein kürzeres und besser lesbares Programm.
- Der Primzahlentest wurde in eine eigene Routine gelegt, die vom Hauptprogramm aus aufgerufen wird.

Starten Sie PRIME1 mit [F9] in Turbo Profiler, damit Sie sich die Statistiken anschauen können. Wählen Sie dann die Option *Display* im lokalen Menü des Fensters *Execution Profile*, um das Dialogfenster *Display Options* zu öffnen. Aktivieren Sie nun die Option *Both*, drücken [Eingabe] und vergrößern das Fenster *Execution Profile* (mit [F5]).

Sie sehen, daß sich die Ausführungszeit etwas verbessert hat (das liegt zum Teil daran, daß PRIME1 weniger Informationen als PRIME0 ausgibt). Den entscheidenden Engpaß bildet immer noch die **printf**-Anweisung (nun in Zeile 21).

Achten Sie vor allem darauf, daß der Primzahlen-Test (Zeile 9 in PRIME0) nun 78022 statt 15122 Male ausgeführt wird. Das sieht auf den ersten Blick gewaltig aus. Bedenken Sie aber, daß sich die Ausführungszeit dadurch nur um eine Sekunde erhöht. Diese Anweisung wird nämlich, wie wir bereits gesehen haben, sehr schnell ausgeführt.

Um eine Verbesserung zu erzielen, müssen wir, da die Testschleife in eine eigene Routine gelegt wurde, ganz offensichtlich versuchen, die Anzahl der Aufrufe für diese Routine zu verringern. Es gibt mehrere Möglichkeiten, um die Anzahl der Integer-Zahlen einzuschränken, die zum Testen an diese Routine übergeben werden müssen. Je mehr sich im Hauptprogramm ausschließen läßt, desto weniger Aufrufe sind nötig und desto schneller wird Ihr Programm ausgeführt. Das ist die Vorgehensweise in den nächsten Beispielprogrammen.

Änderung des Programms und erneute Analyse

Wir haben bereits früher festgestellt, daß nicht alle Faktoren zwischen 1 und n in der Modulo-Anweisung überprüft werden müssen. Es reicht, wenn man die Obergrenze auf die Quadratwurzel der untersuchten Zahl einschränkt. Das geschieht im Programm PRIME2.

Laden des nächsten Programms (PRIME2)

Machen Sie nun weiter und laden Sie PRIME2, die nächste Version des Beispielprogramms, in das Fenster *Module*. Im Programm PRIME2 kommt die Routine **Wurzel** hinzu, die eine Bibliotheksfunktion zur Berechnung der Quadratwurzel aufruft und ein Integer-Ergebnis zurückliefert.

Nun müssen Sie für alle Zeilen des Moduls Bereiche festlegen. Aktivieren Sie das lokale Menü und wählen Sie *Add Areas / Every Line* im *Module*-Fenster. Anschließend drücken Sie die Taste [*Eingabe*].

Drücken Sie [*F9*], um die Analyse zu starten. Wiederum sehen Sie die Ausgabe aller Primzahlen zwischen 1 und 1000 auf dem Bildschirm.

Nach Ablauf des Programms öffnen Sie das Dialogfenster *Display Options* (indem Sie *Display* im lokalen Menü von *Execution Profile* auswählen). Stellen Sie *Display* auf *Both*. Trotz der gesunkenen Anzahl an Aufrufen in Zeile 15 (von 78022 auf 5288) und der kürzeren Zeitdauer für diese Anweisung, ergab sich eine deutliche Erhöhung in der gesamten Ausführungszeit.

Das Problem in PRIME2 ist der Aufwand für die neue Funktion **Wurzel**. Zeile 7 dieser Funktion wird 5456 mal aufgerufen und benötigt die meiste Zeit aller Routinen.

Zeigt das Fenster *Execution Profile* sowohl die Zeiten als auch die Anzahl der Durchläufe an, lohnt es sich, nach charakteristischen Strukturen zu suchen. Weniger effektive Routinen erkennt man daran, daß die zweite Balkenreihe mit den Zeitdaten viel länger ist als die erste Zeile (mit der Anzahl der Aufrufe). Das bedeutet, daß das Verhältnis von Zeit zu Anzahl sehr hoch ist. Dies trifft zum Beispiel für Zeile 27 mit der **printf**-Anweisung zu.

Wenn das Zeit-/Anzahl-Verhältnis einer Routine hoch ist, lohnt es sich, die Routine durch eine andere zu ersetzen.

Die **return**-Anweisung in der Funktion **Wurzel** (Zeile 7) stellt jedoch ein anderes Problem dar. Zeile 7 ist für die größte Anzahl an Aufrufen und die längste Zeitdauer verantwortlich. Zwei andere Zeilen (5 und 8) enthalten 5456 Aufrufe, aber der Balken im Histogramm für die beiden Zeilen deutet auf eine geringe Ausführungsdauer. Dies bedeutet, daß die beiden Anweisungen schnell sind. Das größte Problem ist also die hohe Anzahl an Aufrufen der Funktion **Wurzel**.

Verringerung der Aufrufe einer Funktion (PRIME3)

Die Aufgabe besteht nun darin, die Anzahl der Aufrufe für die **Wurzel**-Funktion zu verringern. Laden Sie PRIME3 in das Fenster *Module*, vergrößern Sie das Fenster *Module* und werfen Sie einen Blick auf den Quellcode.

In PRIME3 wurde lediglich die Funktion **Prim** verändert. Es wurde eine neue Integer-Variable, *Obergrenze*, eingeführt, der vor dem Eintritt in die **for**-Schleife das Ergebnis von **Wurzel(n)** zugewiesen wird. Die **for**-Schleife verwendet die **Obergrenze** als Abbruchbedingung.

Im Fenster *Module* setzen wir für jede Zeile (*Every Line*) des Moduls einen Bereich. Wenn Sie das Programm dieses Mal untersuchen (wählen Sie *Run / Run* oder drücken Sie [F9]), werden Sie feststellen, daß das Programm ein gutes Stück früher mit der Berechnung fertig ist. PRIME3 ist um 25% schneller geworden.

In der **printf**-Funktion findet der wesentliche Zeitverbrauch statt, mehr als die Hälfte der gesamten Ausführungszeit geht zu Lasten dieser Routine. Durch die Verringerung der Anzahl der Aufrufe für die Quadratwurzel-Anweisung in **WurzelD** (von 5456 auf 999) gelang eine wesentliche Verringerung der Berechnungszeit.

Weitere Verbesserungen in der Berechnung (PRIME4)

Es gibt immer noch einige Möglichkeiten, um die Effektivität der **Prim**-Funktion zu verbessern. Laden Sie jetzt PRIME4 in das *Module*-Fenster und sehen Sie sich die Zeilen 8 bis 17 des Quellcodes an.

```

/***** PRIME4.C *****/           ;***** PRIME4PA.PAS *****/

if (n % 2 == 0)
    return (n==2);

if (n % 3 == 0)
    return (n==3);

if (n % 5 == 0)
    return (n==5);

for (i=7; i*i < n; i+=2)
    if (n % i == 0)
        return 0;

return 1;
    
```

Hier wurden eine Reihe von Verbesserungen vorgenommen:

- Die drei **if**-Anweisungen in der Funktion **Prim** schließen Faktoren der Vielfachen von 2, 3 und 5 aus. Wenn Sie eine Zahl **n** anhand einer dieser Tests nicht ausschließen können, müssen Sie alle übrigen Zahlen bis zur Wurzel von **n** prüfen. Sie können mit 7 beginnen – alle Zahlen darunter sind bereits durch die **if**-Anweisungen ausgeschlossen.
- Die **for**-Schleife wird nun in jedem Schleifendurchlauf um 2 erhöht, da gerade Zahlen nicht überprüft werden müssen.

- Die Abfrage $i * i \leq n$ tritt an die Stelle der aufwendigeren Abfrage in der **Wurzel**-Funktion.

Im Endeffekt konnte mehr als eine Sekunde der Ausführungszeit eingespart werden.

CR/LF eliminieren (PRIME6)

Die letzte Änderung: Anstelle der Ausgabe des Zeichenpaars Carriage Return/Linefeed drucken wir lediglich ein Leerzeichen aus. Dies ist die einzige Änderung des Programms PRIME6.

Laden Sie PRIME6, setzen Sie einen eigenen Bereich für jede Zeile, dann starten Sie das Programm.

Welch eine Überraschung! Durch die Entfernung der CR/LF-Zeilenschaltung konnte die Ausführungszeit um den Faktor 7 verkürzt werden. Die Ausgabe in einer neuen Zeile ist scheinbar sehr aufwendig. Die Verteilung der Analysen ist bezüglich der Zeitdauer und Anzahl der Ausführung sehr gleichmäßig. Es ist recht schwierig, das Programm noch weiter zu verbessern, ohne den Algorithmus entscheidend zu verändern.

Wie geht's weiter?

Wir haben in dieser Einführung die Grundlagen für die Analyse von Laufzeiteigenschaften behandelt. Sie sollten mit der Bedienung von Turbo Profiler vertraut sein: Laden und Analysieren von Programmen, Ausgabe verschiedener Fensterinhalte, Speichern und Laden der Profiler-Statistiken, neue Anordnung für die Fenster zur Auswertung der Statistiken.

Beenden Sie nun den Profiler (wählen Sie *File / Quit* oder drücken Sie **[Alt][X]**).

Weitere Informationen über die Benutzeroberfläche von Turbo Profiler und Details zu Teilbereichen, die hier nicht genannt wurden, finden Sie in Kapitel 2.

Einige Anregungen, falls Sie noch weitere Aufgaben bearbeiten wollen:

- Analysieren Sie das Programm mit Primzahlen bis zu:
 - 2500
 - 5000
 - 7500
 - 10000
- Setzen Sie den Profiler-Modus (im Dialogfenster von *Statistics / Profiling*) auf *Passive analysis*. Wie wirkt sich dies auf den Zeitaufwand des Profilers aus? Welche Art von Information geht in der passiven Analyse verloren? (In Kapitel 3 wird die passive Analyse näher erläutert.)

- Stellen Sie fest, wie groß die Verbesserung ist, wenn sie das Sieb des Eratosthenes für die Berechnung der Primzahlen bis 10000 verwenden.
- Vergleichen Sie den Aufwand für die Ausgabe eines Zeilenvorschubs gegenüber einer Cursor-Positionierung.

Hinweis

Es gibt eine Reihe von Artikeln zur Analyse von Laufzeiteigenschaften, aber nur wenig Bücher. Jon Bentleys Buch, *Writing Efficient Programs*, faßt die Regeln für den Entwurf effizienter Programme zusammen, schlägt eine umfassende Methodik für die Analyse vor und beinhaltet eine umfangreiche Sammlung an Literaturhinweisen.

K A P I T E L 2

Die Benutzeroberfläche

Turbo Profiler macht Ihnen die Analyse Ihrer Programme so einfach wie möglich. Beim Aufruf von Turbo Profiler haben Sie sprichwörtlich alles in Ihrer Hand. Alle Befehle und Optionen sind blitzschnell über die Tastatur erreichbar.

Turbo Profiler weist außerdem noch einige Erweiterungen auf, die die Bedienung zusätzlich erleichtern:

- Fenster, die mehrfach überlagert, beliebig verschoben und auf die gewünschte Größe eingestellt werden können;
- Mausunterstützung für jede Maus, die kompatibel zum Microsoft Treiber Version 6.1 ist;
- Dialogfenster anstelle mehrfach verschachtelter Menüs.

Teil 1: Die Bestandteile der Benutzeroberfläche

Es gibt drei sofort ins Auge fallende Komponenten in der integrierten Umgebung: die Menüleiste oben, den Fensterbereich in der Mitte und die Statuszeile unten. Viele Menüoptionen bieten außerdem Dialogfenster. Bevor wir die einzelnen Menüoptionen der Benutzeroberfläche behandeln, beschreiben wir diese allgemeineren Bestandteile.

Die Menüleiste und die einzelnen Menüs

Die Turbo Profiler-Oberfläche enthält sowohl globale als auch lokale Menüs. Globale Menüs sind über die Menüleiste erreichbar. Lokale Menüs sind jeweils einem bestimmten Fenster zugeordnet und sind nur innerhalb dieser Fenster erreichbar.

Die Menüleiste stellt den wichtigsten Zugang zu allen Optionen der globalen Menüs dar. Außerdem wird auf der rechten Seite der Programmstatus angezeigt, der zum Beispiel angibt, ob der Profiler bereit (*READY*) ist, Befehle auszuführen, ob das Programm läuft (*RUNNING*), oder ob der Profiler auf die Beendigung einer aufwendigen Operation wartet (*WAITING*). Die Menüleiste ist immer sichtbar, es sei denn, auf dem Bildschirm werden gerade die Ausgaben Ihres Programms und nicht die Benutzeroberfläche von Turbo Profiler dargestellt.

Die Auswahl von Optionen mit der Tastatur

Um eine Option eines globalen Menüs mit der Tastatur anzuwählen, gehen Sie wie folgt vor:

1. Drücken Sie **[F10]**. Dadurch wird die Menüleiste aktiviert. Nach der Aktivierung der Menüleiste wird einer der Menütitel durch einen Auswahlbalken hervorgehoben. Der hervorgehobene Menütitel zeigt das aktuell gewählte Menü an.
2. Mit den Pfeiltasten und **[Eingabe]**, oder durch die Eingabe des Tastenkürzels, d.h. den hervorgehobenen Buchstaben des gewünschten Menütitels, wählen Sie ein Menü.

Hinweis

Mit **[Esc]** brechen Sie diese Operation ab.

Das Menü *Files* können Sie beispielsweise von der Menüleiste aus mit **[F]**, von jeder beliebigen anderen Stelle aus mit **[Alt][F]** öffnen.

Folgen einer Option des nun geöffneten Menüs drei Punkte (...), wird nach Auswahl dieser Option ein Dialogfenster geöffnet. Wenn einer Option ein Pfeil folgt, verzweigt diese Option zu einem weiteren Menü.

3. Wird durch eine Menüoption ein weiteres Menü geöffnet, wählen Sie die gewünschte Menüoption wiederum mit den Pfeiltasten und **[Eingabe]** aus.

Sie können auch das Tastenkürzel (den hervorgehobenen Buchstaben der Option) in dem jeweiligen Menü eingeben.

Turbo Profiler führt dann entweder einen Befehl aus, oder öffnet ein Dialogfenster oder ein weiteres Menü.

4. Außer den globalen Menüs bietet die Oberfläche von Turbo Profiler innerhalb aktiver Fenster auch noch *lokale Menüs*. Zum Öffnen eines lokalen Menüs drücken Sie innerhalb eines aktiven Fensters **[Alt][F10]**. Weitere Informationen über die lokalen Menüs finden Sie weiter hinten in diesem Kapitel.

Die Auswahl einer Menüoption mit der Maus

Sie verwenden die Maus zur Auswahl von Befehlen in globalen Menüs, indem Sie auf die gewünschte Option in der Menüleiste klicken, um das

Menü anzuzeigen. Klicken Sie dann auf die entsprechende Menüoption. Sie können den Mauszeiger natürlich auch geradlinig zur gewünschten Menüoption ziehen und dann die Maustaste loslassen. (Dieser Vorgang wird abgebrochen, wenn Sie den Mauszeiger aus dem Menü ziehen; es wird dann kein Befehl ausgeführt.)

Tastenkürzel

Turbo Profiler bietet eine Reihe von Möglichkeiten zur schnellen Auswahl von Menübefehlen. Der schnellste Weg für Mausbenutzer sieht folgendermaßen aus: Halten Sie die Maustaste gedrückt, ziehen Sie den Mauszeiger vom Hauptmenü zur gewünschten Menüoption und lassen Sie dann die Maustaste los.

Tastaturbenutzern stehen für denselben Zweck zahlreiche Tastenkürzel zur Verfügung. Sie erreichen bzw. aktivieren Menüoptionen über die in Tabelle 2.1 aufgeführten Tastenkürzel.

Tabelle 2.1:
Tastenkürzel

Tastenkürzel	Funktion
[Strg] und hervorgehobener Buchstabe	Wahl einer lokalen Menüoption.
[Alt] und hervorgehobener Buchstabe	Öffnen eines Menüs aus der Menüleiste.
Hervorgehobener Buchstabe	Wahl einer Option in einem aktiven Menü bzw. eines Eintrags in einem Dialogfenster.
Tastenkürzel-Kombination, die neben einer Menüoption steht.	Ausführen der Menüoption

Fenster in Turbo Profiler

Die Arbeit in der IDE von Turbo Profiler erfolgt fast ausschließlich in Fenstern. Ein Fenster ist ein Teil des Bildschirms, den Sie verschieben, vergrößern und verkleinern, mit anderen Fenstern verdecken, öffnen und schließen können.

Die Anzahl der Fenster, die Sie öffnen können, hängt nur vom verfügbaren Speicher ab – aktiv ist allerdings immer nur ein einziges. Alle Befehle und Texteingaben beziehen sich nur auf dieses Fenster.

Hinweis

Als aktives Fenster wird immer das Fenster bezeichnet, in dem Sie gerade arbeiten.

Zwar gibt es mehrere Fenstervarianten, die meisten davon haben aber folgende Elemente gemeinsam: einen Titelzeile, ein Schließfeld, zwei Bildlaufleisten, ein Eckfeld zur Änderung der Größe, ein Zoom-Feld, ein Symbolfeld und eine Fensternummer (1 - 9).

Verwaltung der Fenster

Einige Fenster sind in zwei oder mehrere Ausschnitte unterteilt, um unterschiedliche Informationen getrennt darzustellen. Einzelnen Ausschnitten ist meist ein eigenes lokales Menü zugeordnet.

In der folgenden Tabelle finden Sie einen Kurzüberblick über die Manipulationsmöglichkeiten von Fenstern in Turbo Profiler: Sie können alle Aktionen sowohl mit der Maus als auch mit der Tastatur durchführen.

Tabelle 2.2:
Fenster manipulieren

Ergebnis	Arbeitsmethoden
Fenster öffnen	Wählen Sie eine der Optionen des Menüs <i>View</i> , um ein Profiler-Fenster zu öffnen.
Fenster schließen	Wählen Sie im <i>Window</i> -Menü <i>Close</i> , drücken Sie [Alt][F3] , oder klicken Sie das Schließfeld an.
Fenster aktivieren	Klicken Sie das Fenster an einer beliebigen Stelle an drücken Sie [Alt] und die Fensternummer (1-9), angezeigt in der rechten oberen Ecke des Fensters, wählen Sie aus der Fensterliste im <i>Window</i> -Menü das gewünschte Fenster aus oder wählen Sie im Menü <i>Windows</i> den Befehl <i>Next</i> (oder drücken Sie [F6]), um das nächste Fenster zu aktivieren.
Fensterinhalt verschieben	Verwenden Sie die Pfeiltasten, um einen Bildlauf nach oben, unten, links oder rechts durchzuführen, oder verwenden Sie die Maus, um die Bildlaufleisten zu bedienen: • Klicken Sie auf einen der Richtungspfeile an den Enden der Bildlaufleisten, um den Fensterausschnitt um eine Zeile oder ein Zeichen in die gewünschte Richtung zu verschieben. • Klicken Sie in den schraffierten Bereich in der Mitte der Bildlaufleisten, um den Fensterausschnitt um eine Fensterlänge in die gewünschte Richtung zu verschieben. • Ziehen Sie die Positionsmarke der Bildlaufleiste so weit in die gewünschte Richtung, daß der entsprechende Fensterausschnitt angezeigt wird.
aktives Fenster verschieben	Klicken Sie mit der Maus den Titel oder ein Rahmenzeichen (nicht aber ein Zoom-Feld, eine Bildlaufleiste, ein Schließfeld oder Symbolfeld) an, und ziehen Sie das Fenster an die gewünschte Position, oder drücken Sie [Strg][F5] (<i>Window/Size/Move</i>) und verwenden Sie die Pfeiltasten zum Positionieren des Fensters. Drücken Sie anschließend [Eingabe] .
Größe des aktiven Fensters ändern	Ziehen Sie mit der Maus das Eckfeld zum Ändern der Größe an die gewünschte Position, oder wählen Sie <i>Size/Move</i> im Menü <i>Window</i> (oder drücken Sie [Strg][F5]), drücken Sie [Umschalt][Pfeiltaste] , um die Größe

	des Fensters zu ändern und dann [<i>Eingabe</i>], oder ziehen Sie jeden Teil des rechten oder unteren Fensterrandes, der keine Titellaufleiste ist.
aktives Fenster vergrößern/verkleinern	Klicken Sie das Zoom-Feld an, oder klicken Sie den Titel zweimal an, oder wählen Sie <i>Window/Zoom</i> oder drücken Sie [<i>F5</i>].
aktives Fenster in Symbol umwandeln	Klicken Sie das Symbolfeld an; oder wählen Sie <i>Window/Iconize/Restore</i> . Wenn ein Fenster in voller Größe dargestellt ist, hat es nur ein Symbolfeld. Wenn es als Symbol (<i>icon</i>) vorliegt, hat es nur ein Zoom-Feld. Im normalen Fensterstatus stehen beide Möglichkeiten zur Verfügung.
Ausschnitt wechseln	Drücken Sie [<i>Tab</i>] oder [<i>Umschalt</i>][<i>Tab</i>] oder [<i>Umschalt</i>][<i>Pfeiltaste</i>] oder wählen Sie <i>Window/Next Pane</i> .

Die Statuszeile

Die Statuszeile am unteren Bildschirmrand von Turbo Profiler enthält folgende Informationen:

- Sie zeigt die grundlegenden Tastenkürzel, die in der aktuellen Situation verfügbar sind. (Sie werden sehen, daß sich die Statusleiste ändert, wenn Sie die Tasten [*Alt*] oder [*Strg*] gedrückt halten.)
- Sie bietet Ihnen Tastenkürzel, die Sie mit einer Maus schnell anklicken können (anstatt eine Option im Menü auszuwählen oder das Tastenkürzel über die Tastatur einzugeben).
- Sie bietet einzeilige Informationen über die ausgewählten Menüoptionen oder Einträge eines Dialogfensters.

Die Statuszeile ändert sich, wenn Sie das aktive Fenster wechseln oder eine andere Operation durchführen. Sie können ein beliebiges Tastenkürzel anklicken, um einen Befehlszuführen.

Die Statuszeile ist nur dann nicht verfügbar, wenn ein Dialogfenster oder ein Menü geöffnet wurde. Sie müssen zuerst das Dialogfenster oder das Menü schließen, bevor Sie etwas anderes machen können.

Wenn Sie eine Menüoption gewählt haben, wird in der Statuszeile ein einzeiliger Hilfstext der jeweiligen Menüoption ausgegeben. Wenn zum Beispiel das Menü *Options* geöffnet ist, zeigt die Statuszeile eine Information über den gerade selektierten Eintrag im Menü *Options* an.

Dialogfenster

Menüoptionen mit Fortsetzungspunkten (...) öffnen ein Dialogfenster, in dem Funktionen ausgeführt und mehrere Einstellungen vorgenommen werden können.

Einstellungen in Dialogfenstern können über sechs verschiedene Steuerungsarten vorgenommen werden: Schaltfelder, Markierungsfelder, Aktionsschalter, Eingabefelder, Listen und Standardfelder.

Bei Farbmonitoren werden die einzelnen Elemente des Dialogfensters in verschiedenen Farben dargestellt.

Teil 2: Die Menüreferenz

Dieser Abschnitt enthält eine Beschreibung aller Menübefehle und Optionen in Dialogfenstern, die in der Benutzerumgebung von Turbo Profiler verfügbar sind.

System-Menü

Das *System*-Menü erscheint als Symbol in der linken oberen Ecke der Menüleiste. Mit [Leerzeichen] wählen Sie das Menü am schnellsten an. Sie können aber auch die Taste [F10] und anschließend die Pfeiltasten dazu verwenden, um das Menüsymbol anzusteuern und anschließend die Taste [Eingabe] drücken.

Mit den Optionen dieses Menüs können Sie:

- den Bildschirm neu aufbauen;
- die ursprüngliche Fenstereinstellung wiederherstellen;
- Informationen über Turbo Profiler anzeigen.

Repaint Desktop

Wählen Sie den Befehl *Repaint Desktop*, wenn Sie den Bildschirm von Turbo Profiler neu ausgeben wollen. Dies kann zum Beispiel dann erforderlich sein, wenn ein speicherresidentes Programm einige Zeichen auf dem Bildschirm nicht gelöscht hat oder wenn Sie die Bildschirmumschaltung ausgeschaltet haben.

Restore Standard

Wenn Sie Turbo Profiler starten, wird die Größe der Fenster, ihr Status (geöffnet oder geschlossen) und die Anordnung auf dem Bildschirm entsprechend den Informationen in der Konfigurationsdatei TFCONFIG.TF eingestellt. Sobald Turbo Profiler geladen ist, können Sie die Fenster verschieben, vergrößern/verkleinern, einige Fenster öffnen oder schließen, und so ein gewaltiges Chaos auf dem Bildschirm produzieren. Der Befehl *Restore Standard* bietet eine schnelle Methode, um eine solche Situation wieder zu bereinigen.

Wenn Sie *Restore Standard* wählen, stellt Turbo Profiler alle Fenster wieder wie zu Beginn der Turbo-Profiler-Sitzung ein.

About

Nach Auswahl der Option *About* erscheint das About-Fenster, in dem die Versionsnummer von Turbo Profiler und andere Informationen stehen. Drücken Sie [*Eingabe*] oder wählen Sie *OK*, um das Informationsfenster zu schließen.

File-Menü

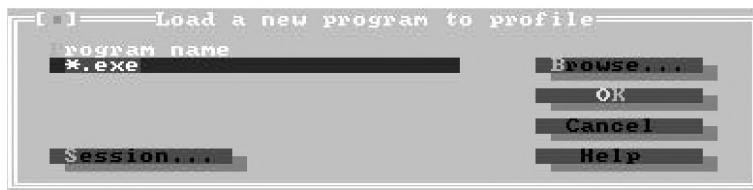
Über das Menü *File* können Sie:

- Programmdateien, die untersucht werden sollen, öffnen und laden,
- das aktuelle Verzeichnis wechseln,
- Informationen über Ihr Programm und die Belegung des Systemspeichers ermitteln,
- in die DOS-Ebene wechseln und
- den Turbo Profiler verlassen.

Open

Der Befehl *File / Open* öffnet das nachfolgend dargestellte Dialogfenster zum Laden eines Programms:

Abbildung 2.1:
Das Dialogfenster
Load A New Program To Profile

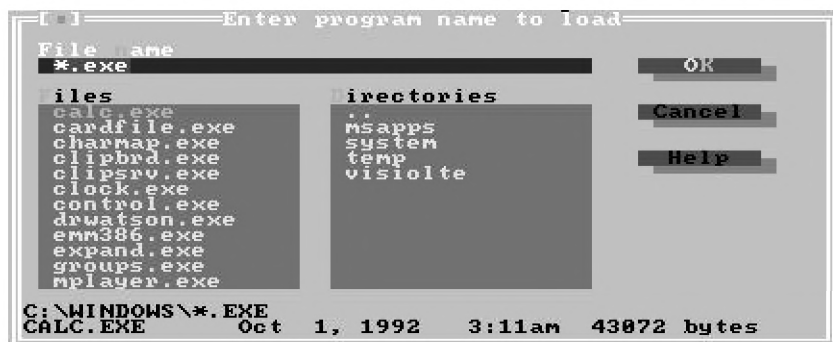


Das Dialogfenster *Load A New Program To Profile* enthält einen zusätzlichen Schalter, *Session*, zur Unterstützung von Remote-Analysen. Weitere Informationen über die Remote-Analyse und den Schalter *Session* finden Sie im Anhang C.

Wenn Sie den Namen des Programms kennen, das Sie laden möchten, geben Sie ihn in das Eingabefeld *Program Name* ein, und drücken Sie [*Eingabe*].

Wenn Sie in verschiedenen Verzeichnissen nach Ihrem Programm suchen möchten, klicken Sie auf den Schalter *Browse*, um ein weiteres Dialogfenster zu öffnen (das Dialogfenster *Enter Program Name To Load*):

Abbildung 2.2:
Das Dialogfenster
Enter Program Name To Load



Das Listenfeld *Files* zeigt alle Dateien im aktuell ausgewählten Verzeichnis an. Durch Eingabe einer Dateimaske in das Eingabefeld *File Name* (wie z.B. *.EXE) können Sie festlegen, welche Dateien aufgelistet werden sollen. Dieses Eingabefeld können Sie auch zum Wechseln des Laufwerks verwenden.

„Um durch die Verzeichnisse zu gehen“, klicken Sie zweimal auf die Einträge, die in der Liste *Directories* aufgeführt sind (der Eintrag ... führt Sie in das nächsthöhere Verzeichnis). Wenn Sie ein Verzeichnis ausgewählt haben, wählen Sie die Datei, die Sie laden möchten, aus dem Listenfeld *Files*. Sie können auch nach einer Datei suchen, indem Sie den Dateinamen in dem Listenfeld *Files* direkt eingeben. Turbo Profiler bewegt den hervorgehobenen Balken auf den Dateinamen, der mit den ersten Zeichen Ihrer Eingabe übereinstimmt. Wenn Sie eine Datei ausgewählt haben, drücken Sie OK. Es erscheint jetzt wieder das Dialogfenster *Load A New Program To Profile*.

TPROF.EXE enthält Schaltfelder in dem Dialogfenster *Load A New Program To Profile*, die die Remote-Analyse unterstützen. Die *Session*-Schaltfelder legen fest, ob sich das Programm, das Sie debuggen möchten, lokal auf Ihrem System oder auf einem Remote-System befindet. Wenn es sich auf einem Remote-System befindet, wählen Sie das Schaltfeld *Remote Windows*; wenn es sich nicht auf einem Remote-System befindet, wählen Sie *Local*. Im Anhang C finden Sie Anweisungen über das Remote-Debuggen.

Hinweis

Bevor Sie ein Programm in den Profiler laden, stellen Sie sicher, dass Sie den Programmcode in eine ausführbare Datei (.EXE oder .DLL) mit Debuggerinformationen kompiliert haben. Obwohl Sie Programme laden können, die keine Debuggerinformationen enthalten, ist es nicht möglich, den Programmcode im Fenster *Module* anzusehen. (Der Profiler kann nicht auf Programmcode referieren, wenn keine Debuggerinformationen vorliegen. Wenn Sie ein Modul laden, das keine Debuggerinformationen enthält, öffnet Turbo Profiler das Fenster *Disassembly*, das die disassemblierten Maschinenbefehle des Moduls enthält.)

Wenn Sie ein Programm unter der Kontrolle von Turbo Profiler laufen lassen möchten, müssen alle ausführbaren Dateien (eingeschlossen alle .DLL-Dateien) und die originalen Quelltextdateien vorliegen. Außerdem müssen sich alle .EXE- und .DLL-Dateien der Anwendung in einem Verzeichnis befinden.

Protokollierungsdatei

Wenn Sie Turbo Profiler beenden, wird eine Protokollierungsdatei in dem aktuellen Verzeichnis erzeugt, die Informationen über den Ablauf der Analyse-Sitzung, die Sie beenden, enthält. Wenn Sie dann erneut Ihr Programm in diesem Verzeichnis laden, stellt Turbo Profiler den Status der letzten Sitzung wieder her.

Standardmäßig werden alle Protokollisten in einer Datei gespeichert. Sie haben den Dateinamen XXXX.TP bei TPROF.EXE bzw. XXXX.TPW bei TPROFW.EXE, wobei XXXX für den Namen des Programms steht, das Sie analysieren. Wenn beim Beenden von Turbo Profiler keine Datei geladen ist, lautet XXXX TPROF bzw. TPROFW.

Der Befehl *Options / Set Restart Options* öffnet das Dialogfenster *Restart Options*, in dem Sie festlegen können, wie Turbo Profiler die Protokollierungsdateien behandeln soll. In diesem Dialogfenster können Sie mit dem Markierungsfeld *Restore At Restart* festlegen, ob die Protokolliste gespeichert werden soll. Die Schaltfelder *Use Restart Info* legen fest, wie die Datei behandelt wird:

Tabelle 2.3:
Speicheroptionen der
Turbo Profiler-
Protokollisten

Option	Beschreibung
Always	Immer die Protokollierungsdatei verwenden.
Ignore if old	Protokollierungsdatei nicht verwenden, wenn Sie Ihr Programm neu kompiliert haben.
Prompt if old	Fragt Sie, ob die Protokollierungsdatei verwendet werden soll, wenn Sie Ihr Programm neu kompiliert haben.
Never	Protokollierungsdatei wird nicht verwendet.

Get Info

Der Befehl *File / Get Info* zeigt Informationen über das untersuchte Programm und die Belegung des Systemspeichers an.

Die Informationen in *Get Info* werden nur angezeigt, die Einstellungen können nicht verändert werden. Die einzelnen Angaben liefern folgende Informationen:

- Program enthält den Namen der untersuchten Datei. Welche Datei analysiert wird, legen Sie mit *File/Open* fest.
- Status gibt den Grund an, warum Turbo Profiler die Programmsteuerung zuletzt erhalten hat: Es kann eine der folgenden Meldungen vorkommen:

Loaded (Programm geladen)
 Control-Break (Unterbrechung durch [Stg] [Untr])
 Terminated, exit code XX (Programm wurde mit Abschlußcode XX beendet)
 Stopped by area (angehalten durch Bereichsoption)
 NMI Interrupt (NMI-Interrupt)
 Exception XX (Ausnahmebedingung XX)
 Divide by zero (Division durch Null)
 No Program loaded (kein Programm geladen)

Mode enthält den Modus der Analyse (aktiv, passiv oder coverage). Die Untersuchungsmethode kann mit den Auswahlfeldern *Profile Mode* im Dialogfenster *Profiling Options* eingestellt werden (das über *Statistics / Profiling Options* zugänglich ist).

- *Collection* sagt aus, ob die Datenerfassung ein- oder ausgeschaltet ist. Die Einstellung für die Datenerfassung erfolgt mit dem Befehl *Statistics/Accumulation*.
- *Memory* ist eingeteilt in
 - *DOS*: Speicher, der von DOS und/oder Gerätetreibern belegt wird.
 - *Profiler*: gesamter Speicher, der für den Profiler verfügbar ist.
 - *Symbols*: Speicher, der von der Symboltabelle des Programms belegt wird.
 - *Program*: Speicher, der von dem Programm belegt wird, das im Moment untersucht wird.
 - *Available*: Größe des restlichen verfügbaren Speichers.
- *DOS version* gibt die in Ihrem System aktive DOS-Version aus.
- Datum und Uhrzeit werden von der Systemuhr übernommen.

Nachdem Sie die Informationen in *Get Info* betrachtet haben, klicken Sie in *OK* und drücken [*Eingabe*], um zum aktuellen Fenster zurückzukehren.

DOS Shell

Mit dem Befehl *File/DOS Shell* können Sie Turbo Profiler temporär verlassen und zur DOS-Ebene wechseln, um einen DOS-Befehl oder ein anderes Programm aufzurufen. Für die Rückkehr zu Turbo Profiler geben Sie nach der DOS-Eingabeaufforderung **EXIT** ein.

Hinweis

Bei Remote-Analysen erscheint die DOS-Zeile auf dem Bildschirm von Turbo Profiler und nicht auf dem Anwenderbildschirm. Damit kann man nach DOS springen, ohne die Programmausgabe zu überschreiben. Da die Ausgabe Ihres Programms immer auf einem der beiden Monitore sichtbar ist, sind die Befehle *Window / User Screen* und [*Alt*][*F5*] ausgeschaltet (In Anhang C finden Sie eine ausführliche Behandlung von Remote-Analysen.).

Quit [*Alt*][*X*]

Mit dem Befehl *File/Quit* wird Turbo Profiler verlassen und aus dem Speicher entfernt. Daraufhin kehrt man zur DOS-Ebene zurück.

Wenn Sie irgendwelche Analysedaten oder Einstellungsparameter beibehalten wollen (wie die Profiler-Statistik, die Analyse- und Anzeigeeoptionen sowie die Anordnung auf dem Bildschirm), können Sie diese Daten mit den Befehlen *Statistics / Save* und *Options / Save* speichern, bevor Sie das Programm verlassen. Sonst gehen die eingestellten Optionen verloren.

Hinweis

Jedesmal, wenn Sie Turbo Profiler verlassen, werden die für das aktuelle Programm eingestellten Bereiche gespeichert; die Einstellungen werden in einer .TFA-Datei gespeichert. Wenn das Programm das nächste mal geladen wird, werden diese Bereichseinstellungen automatisch aktiviert.

View-Menü

Im Menü *View* können Sie mehrere verschiedene Arten von Fenstern öffnen, in denen Informationen zum Laufzeitverhalten Ihres Programms angezeigt werden.

Tabelle 2.4:
Übersicht über die
Profiler-Fenster

Fenster	Beschreibung der Anzeige
Module	Quellcode des untersuchten Programms.
Execution Profile	Statistische Informationen über ein Programm, nachdem das Programm ausgeführt wurde.
Callers	Informationen, wie oft eine Routine von welchen Routinen aufgerufen wurde.
Overlays	Informationen zu Overlays in Turbo Pascal, Turbo C und Turbo Assembler.
Interrupts	Informationen über Interrupt-Aufrufe des Programms.
Files	Informationen über Dateizugriffe.
Areas	Einzelheiten über die Art der Datenerfassung an den markierten Stellen im Quellcode.
Routines	Alle Routinen, die mit einer Bereichsmarkierung versehen werden können.
Disassembly	Der aktuelle Untersuchungsbereich im Module-Fenster als disassemblierter Quellcode.
Text File	Inhalt einer beliebigen Textdatei.
Coverage	Listet standardmäßig die Codeblöcke auf, die nicht ausgeführt worden sind.

Lokale Menüs

Jedes Profiler-Fenster hat sein eigenes lokales Menü (genaugenommen haben einige Fenster mehr als ein lokales Menü; das hängt von der Anzahl der Abschnitte des Fensters ab). Ein lokales Menü enthält Kommandos und Einstellungen, die für den jeweiligen Abschnitt des Fensters typisch sind.

Um ein lokales Menü zu aktivieren, drücken Sie **[Alt][F10]**; gibt es mehrere Fensterabschnitte, können Sie sich mit der Taste **[Tab]** von einem Abschnitt zum nächsten bewegen. Wenn das lokale Menü erscheint, wählen Sie mit den Pfeiltasten das gewünschte Kommando aus und drücken dann **[Eingabe]**, oder Sie drücken gleich den Kernbuchstaben des Befehls. Nachdem Sie ein lokales Kommando ausgewählt haben, führt Turbo

Profilier das Kommando aus, gibt ein Eingabefenster aus oder bringt eine weiteres Menü zur Anzeige.

Um einen Auswahlpunkt aus einem lokalen Menü direkt (d.h. unter Umgehung des lokalen Menüs) aus dem Fenster zur Ausführung zu bringen, geben Sie das entsprechende Tastenkürzel ein; es besteht aus der Taste [Strg] und dem Kennbuchstaben des Auswahlpunkts.

Um das lokale Menü des gerade aktiven Menüs mit der Maus zu aktivieren, betätigen Sie die rechte Maustaste. Anschließend wählen Sie das gewünschte Kommando durch Anklicken des Menüauswahlpunktes aus.

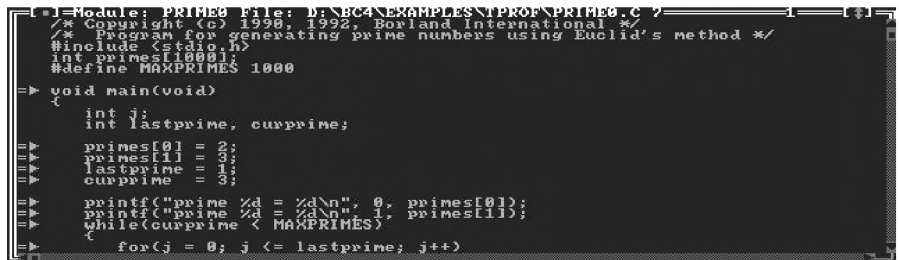
Module

Das Fenster *Module* gibt den Quellcode des Programms aus, das Sie analysieren wollen. Hier können Sie den Code untersuchen und Bereiche festlegen, deren Laufzeitverhalten erfaßt werden soll. Spezielle Tastenkürzel und Verknüpfungen mit anderen Fenstern ermöglichen die gleichzeitige Darstellung des Codes und der Daten zusammen mit den Statistiken in den anderen Fenstern.

Wenn Sie *View/Module* auswählen, erscheint eine Auflistung aller Quellcode-Module, die mit dem gegenwärtig im *Module*-Fenster geladenen Programm gelinkt sind. Steuern Sie das Modul, das Sie im *Module*-Fenster ausgeben lassen wollen, an und drücken Sie OK.

Wenn *Modified* in der Titelzeile des Fensters erscheint, bedeutet dies, daß der Quellcode der Datei, die Sie betrachten, seit der letzten Compilierung des Programms geändert wurde.

Abbildung 2.3:
Das Fenster *Module*
(vergrößert)



```

Module: PRIME0 File: D:\SEC4\EXAMPLES\TPROF\PRIME0.C
/* Copyright (c) 1990, 1992, Borland International */
/* Program for generating prime numbers using Euclid's method */
#include <stdio.h>
int primes[1000];
#define MAXPRIMES 1000
void main(void)
{
    int j;
    int lastprime, curprime;

    primes[0] = 2;
    primes[1] = 3;
    lastprime = 3;
    curprime = 3;

    printf("prime %d = %d\n", 0, primes[0]);
    printf("prime %d = %d\n", 1, primes[1]);
    while(curprime < MAXPRIMES)
    {
        for(j = 0; j <= lastprime; j++)
    
```

Wenn Sie den Profiler aufrufen, benötigen Sie *sowohl* die .EXE-Datei *als auch* die Quelldatei. Turbo Profiler sucht den Quellcode des Programms an den folgenden Stellen in der angegebenen Reihenfolge:

1. Im Verzeichnis, in dem das Programm compiliert wurde. Der Name des Verzeichnisses, in dem das Programm compiliert wurde, ist in der .EXE- und .OBJ-Datei enthalten, *falls* die Übernahme der symbolischen Debuggerinformationen eingeschaltet war);

2. In den Verzeichnissen, die unter *Options / Path* für *Source* angegeben wurden (oder die Sie mit Hilfe der Kommandozeilenoption **-sd** angegeben haben);
3. Im aktuellen Verzeichnis;
4. Im Verzeichnis, in dem sich das untersuchte Programm befindet.

Drücken Sie **[Alt][F10]** oder klicken Sie mit der rechten Maustaste, um das lokale Menü des *Module*-Fensters zu öffnen oder rufen Sie die gewünschte Option direkt auf. Mit den lokalen Menübefehlen können Sie folgende Aktionen durchführen:

- Cursor in eine bestimmte Zeile oder auf ein Label positionieren;
- Suche nach einem bestimmten Text im Quellcode;
- Untersuchungsbereiche hinzufügen oder entfernen;
- Aktion für die Analyse in einem bestimmten Bereich festlegen;
- Ebene der Aufzeichnung des Aufrufpfads für eine bestimmte Routine bestimmen;
- weitere Module oder Quelldateien des aktuellen Moduls in das Fenster *Module* laden;
- einen Editor Ihrer Wahl aufrufen.

Line

[Strg][L]

Um schnell zu einer bestimmten Programmzeile im *Module*-Fenster zu gelangen, wählen Sie den Befehl *Line* des lokalen Menüs aus. Im daraufhin erscheinenden Dialogfenster werden Sie nach der gewünschten Zeilennummer gefragt; geben Sie die neue Zeilennummer ein und wählen Sie dann *OK* aus (oder drücken Sie **[Eingabe]**). Wenn Sie eine Zahl eingeben, die größer ist als die höchste (letzte) Zeilennummer, wird der Cursor an das Dateiende plaziert.

Search

[Strg][S]

Sucht ab der aktuellen Cursor-Position bis zum Dateiende nach einer Zeichenfolge. Es erscheint ein Dialogfenster, in dem Sie den Such-String eingeben. Schließen Sie diese Eingabe mit **[Eingabe]** ab oder betätigen Sie *OK*.

Befindet sich an der Cursorposition ein Text, der möglicherweise ein Variablenbezeichner ist, dann wird dieser Text zur Initialisierung des Dialogfensters *Search* verwendet. Sollten Sie einen Block markiert haben, dann wird dieser Block zur Initialisierung verwendet. Diese Funktion erspart Ihnen das Eintippen der gesuchten Zeichenfolge, wenn der Variablenbezeichner bzw. der markierte Block eben dieser Zeichenfolge entspricht.

Ein Such-String kann einfache Jockerzeichen enthalten; ein **?** steht für nur ein Zeichen, ein ***** für 0 oder mehrere Zeichen.

Die Suche beginnt an der aktuellen Cursorposition und wird nicht am Anfang der Datei fortgesetzt, wenn das Dateiende erreicht wurde. Um die gesamte Datei zu durchsuchen, müssen Sie in der ersten Zeile beginnen.

Next

[Strg][N]

Sucht in der aktuellen Datei nach dem nächsten Vorkommen einer über das Dialogfenster *Search* eingegebenen Zeichenfolge. Dieser Befehl kann daher erst aufgerufen werden, wenn zuvor bereits ein Such-Vorgang durchgeführt wurde. Zum Aufruf des Befehls wählen Sie im lokalen Menü den Befehl *Next* aus, oder drücken Sie [Strg][N].

Goto

[Strg][G]

Um den Cursor auf eine bestimmte Routine oder ein Label im *Module*-Fenster zu positionieren, wählen Sie den Befehl *Goto*. Im daraufhin geöffneten Dialogfenster werden Sie nach der Adresse gefragt, die Sie untersuchen wollen. Geben Sie eine Zeilennummer, einen Routinennamen oder eine Hexadezimaladresse ein und wählen Sie dann *OK*, oder drücken Sie [Eingabe].

Hinweis

Verwenden Sie das jeweilige dezimale Format der aktuellen Programmiersprache.

Weitere Informationen über die Syntax für Adressen finden Sie im *Turbo Debugger Benutzerhandbuch*.

Add Areas

[Strg][A]

Die Option *Add Areas* des Fensters *Module* führt zu einem weiteren Menü mit folgenden Optionen:

- Mit *All Routines* werden alle Routinen des untersuchten Programms, auch die Routinen, für die kein Quellcode vorliegt (wie die Bibliotheks-routinen), für die Analyse markiert.
- *Modules with Source* fügt eine Markierung nur in den Routinen ein, für die der Quellcode zur Verfügung steht.
- *Routines in Module* fügt eine Markierung für alle Routinen des aktuellen Moduls ein (das Modul im Fenster *Module*).
- Der Befehl *Every Line in Module* markiert alle Zeilen des aktuellen Moduls einzeln.
- *Lines in Routine* fügt Bereichsmarkierungen in allen Zeilen der aktuellen Routine ein (der Routine, in der sich der Cursor gerade im Fenster *Module* befindet).
- *Current Routine* fügt eine Bereichsmarkierung in der Routine ein, in der sich der Cursor des Fensters *Module* zu diesem Zeitpunkt befindet.

This markiert die aktuelle Zeile, in der sich der Cursor im *Module*-Fenster befindet, als Bereich.

[Strg][R]

Remove Areas

Die Option *Remove Areas* des Fensters *Module* führt zu einem weiteren Menü. Dieses Menü entspricht größtenteils dem zuvor beschriebenen Menü *Add Areas*. Mit Ausnahme des Befehls *All Areas*, der alle Markierungen entfernt, löscht jeder Befehl des Menüs *Remove Areas* die Bereichsmarkierungen in der gleichen Art und Weise, mit der die entsprechenden *Add Areas*-Befehle Bereichsmarkierungen hinzufügen.

[Strg][O]

Operation

Mit der Option *Operation* wird das Dialogfenster *Area Operation* geöffnet, das die Einstellungen für den aktuellen Bereich enthält (der Bereich, in dem sich der Cursor im Fenster *Module* befindet).

Abbildung 2.4:
Das Dialogfenster
Area Options



Sie können in diesem Dialogfenster zwei Gruppen von Schaltfeldern einstellen: *Operation* und *Timing*. Darüber hinaus ermöglicht das Markierungsfeld *Windows Procedure* das Protokollieren von Meldungen für Windows-Programme.

- Mit *Operation* wird festgelegt, was in diesem Bereich geschehen soll: Wenn Sie einen Bereich markieren, erscheint im Module-Fenster links von diesem Bereich ein Markierungssymbol. Es gibt verschiedene Markierungssymbole: Ein *Normal*-Bereich wird markiert mit =>, ein *Stop*-Bereich mit s>, ein *Enable*-Bereich mit e> und ein *Disable*-Bereich mit d>.
- *Normal* sammelt statistische Daten entsprechend den Vorgaben, die im Statistik-Menü (Aufrufer, Dateiaktivitäten, Interrupts, Overlays usw.) und im Eingabefenster *Area Options* getroffen wurden. Das Dialogfenster *Area Operation* kann sowohl in den lokalen Menüs des Fensters *Module* als auch des Fensters *Areas* aufgerufen werden.
- Mit *Stop* wird die Ausführung des Programms an dieser Markierung angehalten.
- *Enable* schaltet die Erfassung der Statistiken ab dieser Programmstelle wieder ein.
- *Disable* schaltet die Erfassung der Statistiken ab dieser Programmstelle aus. Die Datenerfassung wird wieder aufgenommen, wenn das Programm auf eine *Enable*-Markierung trifft.

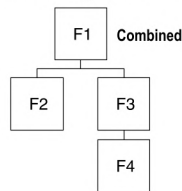
Hinweis

Im *Coverage*-Modus gibt es nur einen Bereichsmarker, den Pfeil. Die Markierung zeigt hier an, daß der bezeichnete Block während des Programmlaufs nicht ausgeführt wurde.

- Das Fenster *Messages* wird aktiv, wenn sich im Markierungsfeld *Windows Procedure* gesetzt wurde. Wenn *Messages* aktiviert wird, erscheint das Eingabefenster *Windows Procedure Messages*.
- In *Timing* wird festgelegt, ob der Profiler die Ausführungszeit aller aufgerufenen Bereiche dem aktuellen Bereich zuordnen oder eine getrennte Erfassung vornehmen soll:
 - Bei *Separate* werden die Zeittakte, die in den aufgerufenen Bereichen stattfinden, ausschließlich dem aufgerufenen Bereich zugeordnet.
 - Mit *Combined* werden die Zeittakte der aufgerufenen Bereiche sowohl der aufrufenden Routine als auch der aufgerufenen zugeordnet. Diese Einstellung ist nur zulässig, wenn die Einstellung des Aufrufers des betreffenden Bereichs auf *Immediate* oder *All* steht.

Wird *Timing* einer Routine auf *Combined* eingestellt, zeigt Turbo Profiler die Zeitstatistiken von Unterroutinen der Routine nicht an – deren Zeiten werden als Teil von der Routine erfasst, deren *Timing* auf *Combined* eingestellt ist.

Die folgende Abbildung zeigt, wie *Timing* funktioniert:



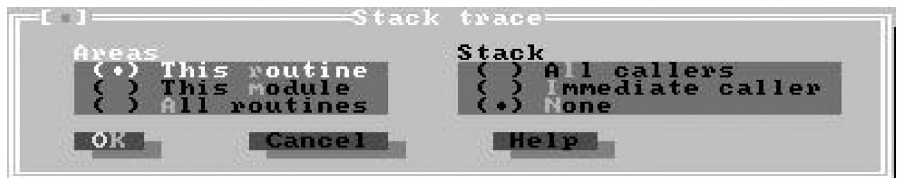
In dieser Abbildung ist *Timing* für die Routine F1 auf *Combined* eingestellt. Turbo Profiler sammelt die Zeitinformationen aller Routinen (F1, F2, F3 und F4) und summiert sie zu einer gesamten Zeitinformation für die Routine F1.

Callers

[Strg][C]

Abbildung 2.5:
Das Dialogfenster
Stack Trace

Der Befehl *Callers* des lokalen Menüs führt in das Dialogfenster *Stack Trace*.



Sie können zwei Gruppen von Schaltfeldern, *Areas* und *Stack*, in diesem Dialogfenster einstellen.

- In *Areas* wird festgelegt, für welche Bereiche ein Aufrufpfad aufgezeichnet werden soll:

- *This Routine* zeichnet alle Aufrufinformationen für die aktuelle Routine auf (das ist die, in der im Module-Fenster der Cursor steht) entsprechend den im Stack-Fenster getroffenen Einstellungen behandelt.
- *This Module* zeichnet alle Aufrufinformationen für das aktuelle Modul auf.
- *All Routines* zeichnet die Aufrufinformationen für alle Routinen in allen Modulen des Programms auf.
- Das Auswahlfeld *Stack* gibt an, wie „tief“ der Aufrufstack verfolgt werden soll:
 - *All Callers* zeichnet alle verfügbaren Aufrufinformationen für die Routine(n) auf, die unter *Areas* angegeben wurde(n).
 - *Immediate Callers* hält in den Aufrufinformationen nur die direkt aufrufende(n) Routine(n) fest, die unter *Areas* angegeben wurde(n).
 - Mit *None* werden keine Aufrufinformationen für die unter *Areas* angegebene(n) Routine(n) protokolliert.

Wenn im Dialogfenster *Stack Trace* das Schaltfeld *OK* gewählt wird, werden die in der Option *Areas* genannten Bereiche entsprechend der geltenden Stack-Option behandelt. Änderungen, die Sie in diesem Eingabefenster durchführen, spiegeln sich im *Areas*-Fenster wider.

Module

[Strg][M]

Die Option *Module* des lokalen Menüs öffnet das Dialogfenster *Pick a Module*, in dem alle Module des Programms aufgelistet sind, für die ein Quellcode zur Verfügung steht.

Die meisten Module bestehen nur aus einer einzigen Quelldatei; in den eingliederten Dateien des Moduls (wie die Deklarationsdateien von C) werden normalerweise nur Konstanten und Datenstrukturen definiert. Verwenden Sie diese Option, um ein anderes Modul im *Module*-Fenster zu öffnen.

Im Rahmen dieser Option werden nur Module angegeben, für die ein Quellcode vorliegt. Außerdem müssen sie zum untersuchten Programm gehören. Damit können Sie schnell von einem Modul in ein anderes wechseln, ohne daß Sie das/die Verzeichnis(se) mit den Quelldateien extra durchsuchen müssen.

Durch die Option *Module* werden die Quelldateien in der nachfolgend angegebenen Reihenfolge durchsucht:

1. im Verzeichnis, in dem sich die Quelldatei während der Compilierung befand;
2. in den Verzeichnissen, die (evtl.) unter *Options / Path for Source* (oder unter Verwendung der Kommandozeilenoption *-sd*) angegeben wurden;
3. im Verzeichnis des Programms, das Sie gerade analysieren;

4. im aktuellen Verzeichnis.

File

[Strg][F]

Der Befehl *File* des lokalen Menüs öffnet ein Dialogfenster, in dem alle Quelldateien aufgelistet sind, die zur Compilierung des aktuellen Moduls verwendet wurden. Verwenden Sie diesen Befehl, wenn sich der Quellcode Ihres Moduls aus mehreren Quelldateien zusammensetzt und die gewünschte Datei momentan nicht im *Module*-Fenster dargestellt wird.

Der Befehl *File* sucht nach den Quelldateien in der gleichen Reihenfolge wie der Befehl *Module*.

Edit

[Strg][E]

Turbo Profiler hat zwar keinen integrierten Editor, Sie können aber den Editor Ihrer Wahl bei der Installation des Profilers mit dem Programm TFINST, angeben. Weitere Informationen über TPROF finden Sie im Anhang B.

Wenn Sie bei der Installation einen Editor angegeben haben, dann wird dieser über den Befehl *Edit* des lokalen Menüs des *Module*-Fensters aufgerufen. Nach Verlassen des Editors übernimmt Turbo Profiler automatisch wieder die Steuerung.

Execution Profile

Im Fenster *Execution Profile* stellt Turbo Profiler die Meßergebnisse der Programmanalyse dar (nachdem Sie Bereiche angegeben haben und das Programm im Profiler gestartet haben).

Execution Profile hat einen Ausschnitt, der in zwei Ausgabebereiche unterteilt ist (oben und unten). Der obere Ausgabebereich enthält:

- *Total Time*: die gesamte Ausführungszeit Ihres Programms.
- *% of Total*: Anteil (als Prozentsatz) der im unteren Bereich dargestellten Meßdaten an der gesamten Ausführungsdauer.
- *Runs*: den aktuellen Analyselauf (wenn Sie mehr als einen Programmlauf erfassen und daraus den Durchschnitt bilden).
- die Optionen, die in den lokalen Menüs ausgewählt wurden (Ausgabeformat, Filterstatus, Sortierreihenfolge).
- *Total Ticks*: die gesamte Anzahl von Zeittakten während eines Programmlaufs. Zeittakte werden nur bei Analysen im passiven Modus angezeigt.

Der untere Bereich enthält ein oder zwei Datenzeilen für jeden markierten Bereich. Zu den Informationen in diesem Anzeigebereich zählen der Name des Bereichs oder die Zeilennummer, die Anzahl der Ausführungen in den markierten Bereichen, die Zeit, die in den markierten Bereichen zugebracht wurde, die durchschnittliche Zeitdauer pro Durchlauf der markierten Bereiche und die maximale Zeitdauer, in der das Programm bei einem bestimmten Durchlauf in einem markierten Bereich war.

Wenn Sie ein *Module*-Fenster und das Fenster *Execution Profile* gleichzeitig auf dem Bildschirm haben, wird das Fenster *Execution Profile* automatisch so positioniert, daß immer die Statistik des Bereichs angezeigt wird, in dem sich der Cursor im *Module*-Fenster befindet.

Um festzulegen, wie die Meßdaten im Fenster *Execution Profile* dargestellt werden, aktivieren Sie das lokale Menü (drücken Sie dazu [Alt][F10]). Mit diesem lokalen Menü können Sie:

- auswählen, welche Art von Modul analysiert werden soll: Windows-Prozeduren oder normale Bereiche;
- eine von sechs Methoden für die Darstellung der Profiler-Statistiken im Fenster *Execution Profile* auswählen;
- die Meßdaten sortieren;
- temporär die Meßdaten eines oder mehrerer Bereiche von der Anzeige entfernen;
- den Quellcode eines Bereichs untersuchen;
- die Meßdaten eines Bereichs aus dem Speicher löschen.

Display

[Strg][D]

Wenn Sie die Option *Display* im lokalen Menü von *Execution Profile* wählen, erscheint das Dialogfenster *Display Options*.

Abbildung 2.6:
Das Dialogfenster
Display Options



In diesem Dialogfenster können Sie drei Gruppen von Schaltfeldern einstellen, *Profile*, *Display* und *Sort*.

- *Profile* legt fest, welche Bereiche im Fenster *Execution Profile* dargestellt werden:
 - *Normal Areas* gibt alle markierten Bereiche aus, darunter auch alle markierten Windows-Prozeduren.
 - *Window Procs* filtert die Anzahl der auszugebenden Bereiche in der Weise, daß nur die Klassen und Windows-Meldungen ausgegeben werden, die im Eingabefenster *Windows Procedure Messages* dazu bestimmt wurden.
Eine vollständige Beschreibung des Dialogfensters *Windows Procedure Messages* finden Sie in Anhang D.
- *Display* bestimmt, welche Daten angezeigt werden:
 - Durch *Time* werden die Meßdaten für die Bereiche als Zeitdauer (in Millisekunden) angegeben, in der sich das Programm in diesem Bereich aufgehalten hat.

- Mit *Counts* wird die Statistik als Anzahl von Durchläufen angezeigt – wie oft das Programm in diesen Bereich gekommen ist.
- *Both* kombiniert die Ausgabe der Zeit (obere Zeile) mit der Anzahl. Dadurch ergibt sich eine grafische Veranschaulichung der Effektivität einer Routine.
- Mit Hilfe von *Per Call* wird das Verhältnis von Zeit/Anzahl angegeben. Dieser Darstellung entnehmen Sie die durchschnittliche Dauer für die Ausführung einer Routine.
- *Longest* zeigt für jeden Bereich die längste Zeitdauer, die das Programm beim Eintritt in einen Bereich zugebracht hat.
- *Modules* gibt für jedes Modul die Zeit aus, in der das Programm in dem Modul verweilt. Das Aktivieren dieser Option ist nur im passiven Modus von Turbo Profiler sinnvoll, d.h. mit jeder Zeitnahme wird aufgezeichnet, welches Modul von der Programmsteuerung gerade ausgeführt wird. Die Option *Modules* verhilft Ihnen bei der Analyse großer Programme zu einem ersten Überblick.
- *Sort* gibt an, in welcher Reihenfolge die Daten sortiert werden:
 - *Name* sorgt dafür, daß die Meßdaten in alphabetischer Reihenfolge der Bereichsnamen aufgelistet werden.
 - *Address* sortiert die Statistik nach aufsteigenden Speicheradressen.
 - Mit *Frequency* werden die Bereiche mit der höchsten Frequenz zuerst angegeben.

Die aktuelle Einstellung der Anzeige- und Sortieroptionen können Sie jederzeit dem oberen Anzeigebereich von *Execution Profile* entnehmen.

Filter

[Strg][F]

Nach Auswahl des lokalen Menübefehls *Filter* erhalten Sie folgendes Menü mit drei Einträgen:

- *All* stellt alle erfaßten Statistiken des aktuellen Programms im Fenster *Execution Profile* dar.

Nachdem Sie bestimmte Statistiken (mit *Filter / Module* oder *Filter / Current*) von der Anzeige in *Execution Profile* ausgeschlossen haben, wählen Sie *Filter/ All* aus, um wieder alle Profiler-Daten im Fenster anzuzeigen.

- Mit *Module* können alle Meßdaten, mit Ausnahme der Daten des nachfolgend angegebenen Moduls, herausgefiltert werden.

Mit dieser Option erhalten Sie das Dialogfenster *Pick a Module*, in dem alle Module des aktuellen Programms aufgelistet werden. Verwenden Sie die Pfeiltasten [*Nach oben*] und [*Nach unten*], um ein Modul in der Liste mit dem Auswahlbalken hervorzuheben, und drücken Sie dann [*Eingabe*]. Im Fenster *Execution Profile* erscheinen nur noch die Bereiche des ausgewählten Moduls.

- Durch die Option *Current* lassen sich die Meßdaten der aktuell ausgewählten Routine vom Fenster *Execution Profile* entfernen.

Wählen Sie *Filter/Current*, wenn Sie die Meßdaten einer bestimmten Routine weglassen wollen, damit Sie die prozentuale Verteilung der restlichen Routinen sehen. Der Befehl *Current* stellt eine vorübergehende Auswahl dar, mit der bestimmte Informationen versteckt werden können, ohne daß Informationen gelöscht werden; dabei geschieht folgendes:

1. Zunächst werden die Meßdaten des aktuellen Bereichs aus *Execution Profile* entfernt.
2. Dann wird die Zeitdauer des entfernten Bereichs von der gesamten Ausführungszeit abgezogen.
3. Anschließend werden die Prozentsätze der übrigen Bereiche als Bruchteil der gesamten Ausführungszeit Neuberechnet.

Wenn die Meßdaten von einem oder mehreren Bereichen entfernt werden, berechnet der Profiler die neue Gesamtzeit für die Ausführung auf der Basis der Meßdaten im Fenster. Der im oberen Bereich des Fensters angezeigte Gesamtwert der Ausführungszeit wird aber nicht verändert.

Wenn Sie den Befehl *Filter/Current* verwenden, bleibt die Anzeige der ursprünglichen Ausführungszeit für das gesamte Programm im oberen Bereich erhalten.

Filter/Current ist ein temporärer Filter, der nur die Report-Informationen verdeckt; dagegen werden mit *Remove* die Bereichsmarkierungen in den Fenstern *Module* und *Areas* für immer entfernt.

Hinweis

Verwechseln Sie *Filter/Current* nicht mit dem ebenfalls im lokalen Menü von *Execution Profile* stehenden Befehl *Remove*.

Module

[Strg][M]

Mit dem lokalen Menübefehl *Module* gelangen Sie in die Zeile des Quellcodes, dessen Meßdaten im Fenster *Execution Profile* ausgewählt sind. Außer mit dem Tastenkürzel [Strg][M] können Sie dieses Kommando auch noch vom Fenster *Execution Profile* aus mit der [Eingabe]-Taste ausführen lassen.

Wenn Sie zum Beispiel die Statistiken der Routine **frank** im Fenster *Execution Profile* selektieren, aktivieren Sie durch die Auswahl des Befehls *Module* im lokalen Menü eine Verbindung. Turbo Profiler aktiviert das *Module*-Fenster und stellt den Cursor in die erste Zeile des Quellcodes von **frank**. Anschließend bewegen Sie die Cursor in Zeile 25 des *Module*-Fensters (in Zeile 25 sei eine neue Bereichsmarkierung enthalten). Daraufhin wird automatisch der Inhalt des Fensters *Execution Profile* so verschoben, daß Sie nun die Meßdaten für Zeile 25 an der Spitze des Darstellungsbereichs sehen.

Die Verbindung erfolgt nur in eine Richtung: Wenn Sie wieder in das Fenster *Execution Profile* (vom Fenster *Module*) wechseln, und den Auswahlbalken verschieben, wird die neue Position im *Module*-Fenster nicht berücksichtigt. (Dies wäre auch nicht wünschenswert.)

Wenn Sie den Befehl *Module* im lokalen Menü von *Execution Profile* auswählen und kein Quellcode für die hervorgehobene Zeile verfügbar ist, wird eine Verbindung zur entsprechenden Code-Zeile im Fenster *Disassembly* (CPU) hergestellt. Dies kann zum Beispiel dann passieren, wenn Sie Bereiche mit *All Routines* markiert haben und die hervorgehobene Zeile eine Bibliotheksroutine enthält (Einzelheiten zu dem Fenster *Disassembly* finden Sie weiter hinten in diesem Kapitel).

Position

[Strg][P]

Das Kommando *Position* entspricht dem Kommando *Module* - mit dem Unterschied, daß das *Position*-Kommando das *Module*-Fenster nicht aktiviert; vielmehr bleibt das Fenster *Execution Profile* aktiv. Eine andere Kurzform für dieses Kommando ist [Leer].

Remove

[Strg][R]

Der Befehl *Remove* entfernt die Bereichsmarkierung in der aktuellen Zeile in den Fenstern *Module* und *Areas*.

Achtung

Der Befehl *Remove* löscht die statistischen Daten. Verwenden Sie diesen Befehl deshalb nur mit Vorsicht.

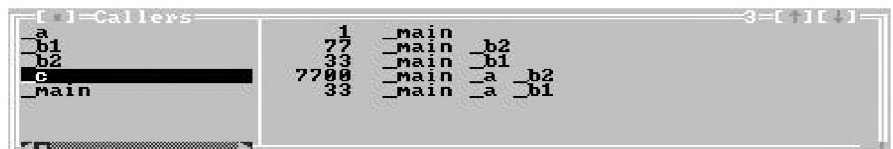
Wenn Sie die Bereichsmarkierungen einer Zeile mit dem Befehl *Remove* entfernen, werden die Meßdaten dieser Zeile gelöscht und keine weiteren Daten mehr für diese Zeile erfaßt. Um einen *Remove*-Befehl rückgängig zu machen, müssen Sie wie folgt vorgehen:

1. Aktivieren Sie das lokale Menü im *Module*-Fenster.
2. Bewegen Sie den Cursor in die Zeile, deren Markierung entfernt wurde.
3. Wählen Sie den Befehl *Add Areas/This line* aus.
4. Starten Sie das Programm erneut (um neue Meßdaten zu erfassen).

Callers

Im Fenster *Callers* stellt Turbo Profiler die Aufrufpfade für alle entsprechend markierten Routinen in Ihrem Programm dar. Ein Aufrufpfad ist eine Liste aller Routinen, die durch die aktuell markierte Routine aufgerufen werden. Sie müssen die Menüoption *Statistics/Callers* auf *Enabled* einstellen, damit vom Profiler Informationen über die Aufrufhierarchie erfaßt werden.

Abbildung 2.7:
Das Fenster *Callers*
mit Aufrufen aus
CALLTEST



Im linken Ausschnitt des *Callers*-Fensters werden die einzelnen Routinen nach Namen alphabetisch sortiert aufgeführt. Wenn Sie eine Routine im linken Ausschnitt auswählen, werden im rechten Ausschnitt die entsprechenden Aufrufpfade für diese Routine angezeigt. Wenn ein Aufrufpfad so lang ist, daß er über den Rand des rechten Ausschnitts hinausreicht, können Sie das Fenster vergrößern, oder ins rechte Fenster überwechseln und den Pfad nach rechts und links rollen.

Hinweis

Vor den Bezeichnern steht im *Callers*-Fenster ein Unterstrich, da Borland C++ allen Symbolen, die in .OBJ-Dateien und in den symbolischen Debuggerinformationen enthalten sind, einen Unterstrich voranstellt.

Auch wenn im *Callers*-Fenster alle Aufrufinformationen *angezeigt* werden, müssen die entsprechenden Optionen für die Erfassung der Aufrufe entweder im *Module*- oder *Areas*-Fenster *angegeben* werden.

Im *Module*-Fenster können Sie die Aufrufoptionen für eine ganze Gruppe von Routinen festlegen:

1. Drücken Sie [Alt][F10] für das lokale Menü, wenn Sie sich mit dem Cursor im *Module*-Fenster über einer markierten Routine befinden.
2. Durch die Auswahl von *Callers* aktivieren Sie das Dialogfenster *Stack Trace*.
3. Stellen Sie die Option *Areas* ein. Sie können wählen, ob alle Aufrufpfade für die aktuelle Routine, alle Routinen des aktuellen Moduls, oder alle Routinen im Programm (mit allen Bibliotheksfunktionen) erfaßt werden sollen.
4. Stellen Sie anschließend die Option *Stack* ein. Sie haben die Wahl, alle aufrufenden Routinen für die gewählte Routine oder nur die Routinen auszuwählen, die in der Aufrufhierarchie direkt über der ausgewählten Routine stehen, oder gar keine Routinen zu erfassen.
5. Drücken Sie [-] oder wählen Sie OK, um wieder zum Fenster *Module* zurückzukehren.

Im Fenster *Areas* können Sie die *Callers*-Optionen für einzelne markierte Routinen angeben (siehe *Areas* für weitere Informationen zu diesem Fenster):

1. Drücken Sie [Alt][F10] für das lokale Menü, wenn Sie sich mit dem Cursor im Fenster *Areas* über einer markierten Routine befinden.
2. Durch die Auswahl von *Options* aktivieren Sie das Dialogfenster *Area Options*.
3. Stellen Sie anschließend die Option *Callers* ein. Sie haben die Wahl, alle aufrufenden Routinen für die gewählte Routine oder nur die Routinen auszuwählen, die in der Aufrufhierarchie direkt über der ausgewählten Routine stehen, oder gar keine Routinen zu erfassen.

4. Drücken Sie *[~]* oder wählen Sie *OK*, um wieder zum Fenster *Areas* zurückzukehren.

In Abbildung 2.7 ist die Routine **c** im linken Ausschnitt des *Callers*-Fensters dargestellt, nachdem das Programm **CALLTEST** analysiert wurde.

```
/* Program CALLTEST */
/* Copyright (c) 1990, Borland International */
#include <stdio.h>

main()
{
    c();
    b2();
    b1();
    a();
}

a()
{
    int i;

    for (i=0; i < 100; i++)
        b2();
    b1();
}

b1()
{
    int i;

    for (i=0; i < 33; i++)
        c();
}

b2()
{
    int i;

    for (i=0; i < 77; i++)
        c();
}

c()
{
    int i;

    for (i=0; i < 3; i++)
        ;
}
```

Der rechte Ausschnitt im Fenster *Callers* listet alle unterschiedlichen Aufrufpfade der Routine **c**:

- 1 Aufruf von **main** an **c**;
- 77 Aufrufe von **main** über **b2** an **c**;
- 33 Aufrufe von **main** über **b1** an **c**;
- 7700 Aufrufe von **main** über **a** und **b2** an **c**;
- 33 Aufrufe von **main** über **a** und **b1** nach **c**.

Sie werden feststellen, daß das Fenster *Callers* sehr praktisch ist, wenn Sie Entscheidungen über eine eventuelle Umstrukturierung des Codes treffen sollen – vor allem dann, wenn eine Routine über mehrere verschiedene Pfade aufgerufen wird.

Die beiden Ausschnitte des Fenster *Callers* haben jeweils ein eigenes lokales Menü. Im rechten Ausschnitt des Fensters *Callers* werden mit beiden Menüoptionen, wie nachfolgend dargestellt, weitere Menüs aufgerufen.

Inspect (linker Ausschnitt)

[Strg][I]

Wenn sich der Auswahlbalken im linken Ausschnitt über der gewünschten Routine befindet, wählen Sie *Inspect* (oder drücken [Strg][I]), um den Quellcode dieser Routine im Fenster *Module* anzuzeigen.

Inspect (rechter Ausschnitt)

[Strg][I]

Befindet sich der Auswahlbalken über einem Aufrufpfad im rechten Ausschnitt des Fensters *Callers*, können Sie die Bestandteile dieses Aufrufpfads in einem von drei weiteren Fenstern untersuchen:

1. Wählen Sie *Inspect*, um eine Liste mit den anderen Fenstern zu erhalten.
2. Wählen Sie das gewünschte Fenster (*Areas*, *Module* oder *Profile*) aus der Liste. Damit erhalten Sie das Dialogfenster *Pick a Caller*, in dem alle aufrufenden Routinen des aktuellen Aufrufpfads aufgelistet werden.
3. Im Dialogfenster selektieren Sie die gewünschte Routine (mit Hilfe der Pfeiltasten oder der Maus) und wählen OK oder drücken [Eingabe]. Wenn das gewünschte Fenster noch nicht geöffnet ist, wird es vom Profiler automatisch geöffnet und zeigt den Programmbereich mit der aufrufenden Routine an.

Sort (rechter Ausschnitt)

[Strg][S]

Mit Hilfe des Befehls *Sort* lassen sich die aufrufenden Routinen im rechten Fenster nach zwei Kriterien sortieren:

- Mit *Called* werden die Aufrufpfade in der gleichen Reihenfolge sortiert, in der Sie im Programm zur Laufzeit aufgerufen wurden.
- *Frequency* sortiert die Aufrufpfade nach der Häufigkeit, in der das Programm einen bestimmten Aufrufpfad benutzte, wobei der am häufigsten vorkommende Aufrufpfad am Anfang steht.

Overlays

Im *Overlays*-Fenster werden Informationen über die Overlay-Zugriffe in Programmen zusammengefaßt, die in Borland C++ oder Turbo Assembler geschrieben wurden. Sie müssen die Menüoption *Statistics/Overlays* auf *Enabled* eingestellt haben, damit Turbo Profiler Overlay-Zugriffe erfassen kann (wenn ihr Programm Overlays hat, aktiviert Turbo Profiler automatisch die Option *Overlays*).

Abbildung 2.8:
Das Fenster *Overlay*



In diesem Fenster wird angezeigt:

- wie oft Ihr Programm die einzelnen Overlays in den Speicher geladen hat;
- wie lange es gedauert hat, ein Overlay zu laden;
- in welcher Reihenfolge die Overlays geladen wurden;
- die Größe der Überlappung bei den Overlays.

Wie *Execution Profile* ist auch das Fenster *Overlays* in zwei Anzeigebereiche – oben und unten – unterteilt. Der obere Anzeigebereich zeigt die Ausführungsdauer für das gesamte Programm und die aktuelle Einstellung für die Anzeige der Overlay-Statistiken. Der untere Anzeigebereich stellt die Overlay-Statistik entweder als Histogramm (*Count*) oder als chronologische Auflistung (*History*) der Overlay-Zugriffe dar.

Wenn Sie einen der Borland Pascal Compiler besitzen, können Sie mit dem Programm OVRDEMO sehen, wie das Fenster *Overlay* funktioniert. Laden Sie dieses Programm in den Profiler. Markieren Sie dann alle Zeilen im Modul OVRDEMO als einen gesamten Bereich, aktivieren Sie *Statistics/Overlays* mit der Option *Enabled* und starten Sie das Programm. (Sie benötigen außerdem die Dateien OVRDEMO.PAS, OVRDEMO1.PAS, OVRDEMO2.PAS und OVRDEMO.EXE, um eine Analyse des Programms durchführen zu können.)

Hinweis

Die Ausführung des Programms OVRDEMO kann durch eine beliebige Taste angehalten werden.

Das lokale Menü von *Overlays* enthält zwei Optionen: *Display* und *Inspect*.

Display

[Strg][D]

Mit *Display* legen Sie die Art der Darstellung für die Daten fest. Drücken Sie [Eingabe], um zwischen *Count* und *History* umzuschalten.

Mit *Count* wird ein Histogramm in der oben dargestellten Art und Weise aus den Daten der Overlays erzeugt. Sie sehen, wieviel Speicher vom

Overlay belegt wird und wie oft das Overlay vom Programm in den Speicher geladen wurde.

Mit *History* werden die Overlay-Zugriffe chronologisch dargestellt. In einer Zeile steht der Name des Overlays und der Zeitpunkt, zu dem das Overlay geladen wurde.

Inspect

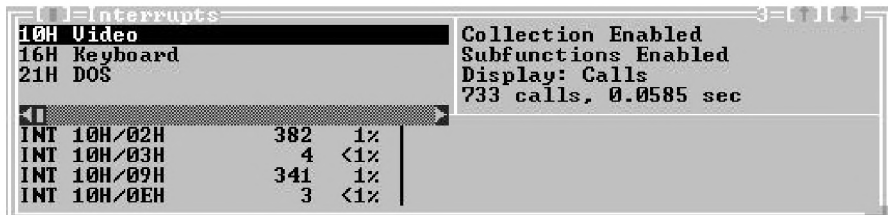
Mit *Inspect* kommen Sie direkt in das *Module*-Fenster (es wird gegebenenfalls neu geöffnet) an die Stelle mit dem Quellcode des ausgewählten Overlays.

[Strg][I]

Interrupts

Im Fenster *Interrupts* finden Sie Informationen über die Interrupts von Bildschirm, Disk, Tastatur, DOS und Maus in Ihrem Programm. Die Menüoption *Statistics/Interrupts* muß auf *Enabled* eingestellt sein, damit der Profiler die Informationen zu den Interrupt-Aufrufen erfaßt.

Abbildung 2.9:
Das Fenster
Interrupts



Das Fenster *Interrupts* ist in drei Ausschnitte – links oben, rechts oben sowie unten – unterteilt.

- Der Ausschnitt links oben enthält die einzelnen Interrupts, die von Ihrem Programm aufgerufen wurden (mit Nummer und Bezeichnung).
- Der Ausschnitt rechts oben enthält Informationen darüber, welche Anzeige im unteren Ausschnitt aktiv ist, und den aktuellen Interrupt, die Anzahl der Aufrufe sowie die Ausführungszeit. In dieses Fenster können Sie nicht mit [Tab] wechseln. Es dient lediglich zur Anzeige von Informationen.
- Im unteren Ausschnitt sehen Sie die Meßdaten jedes einzelnen Interrupts, die entweder als Histogramm oder als Zeitpunkt des Aufrufs zusammen mit der Dauer des Aufrufs dargestellt werden.

Jeder Eintrag des unteren Ausschnitts kann folgende Daten enthalten:

- die Nummer und/oder den Namen des Interrupts;
- die Anzahl der Aufrufe dieses Interrupts (in absoluten Zahlen oder prozentual);
- den gesamten Zeitraum, den das Programm in diesem Interrupt zubachte (als Absolutwert oder in Prozenten).

Beide Ausschnitte des Fensters *Interrupts* haben lokale Menüs.

[Strg][C]	Collection (oberer Ausschnitt) Mit dem Befehl <i>Collection</i> wird die Erfassung von Meßdaten für den aktuellen Interrupt (der im linken Bereich des oberen Ausschnitts hervorgehoben ist) ein- oder ausgeschaltet.
[Strg][S]	Subroutine (oberer Ausschnitt) Der Befehl <i>Subroutines</i> schaltet die Erfassung von Statistiken für bestimmte Unterrouinen des aktuellen Interrupts ein oder aus (dies ist insbesondere für den DOS-Interrupt 21h interessant). Die Nummer der Unteroutine wird dem Wert des AH-Registers beim Aufruf des Interrupts entnommen.
[Strg][A]	Add (oberer Ausschnitt) Mit dem Befehl <i>Add</i> wird ein Interrupt mit einer bestimmten Nummer in die Liste des linken oberen Ausschnitts eingefügt. (Die Nummer des Interrupts wird dezimal eingegeben: Schreiben Sie 33 anstatt 21h.)
[Strg][P]	Pick (oberer Ausschnitt) Mit dem Befehl <i>Pick</i> können Sie sich eine Liste von Interrupts anzeigen lassen, aus der Sie den gewünschten Interrupt für den linken oberen Ausschnitt auswählen können.
[Strg][R]	Remove (oberer Ausschnitt) Mit dem Befehl <i>Remove</i> wird der aktuell ausgewählte Interrupt aus der Liste im linken oberen Ausschnitt entfernt.
[Strg][D]	Delete All (oberer Ausschnitt) Mit dem Befehl <i>Delete All</i> werden alle aufgeführten Interrupts des linken oberen Ausschnitts entfernt.
[Strg][D]	Display (unterer Ausschnitt) Das lokale Menü des unteren Ausschnitts im Fenster <i>Interrrupts</i> hat nur einen Eintrag, <i>Display</i>, der zu einem weiteren Menü führt. In diesem zweiten Menü können Sie auswählen, in welchem von vier Formaten die Interrupt-Statistiken angezeigt werden sollen.

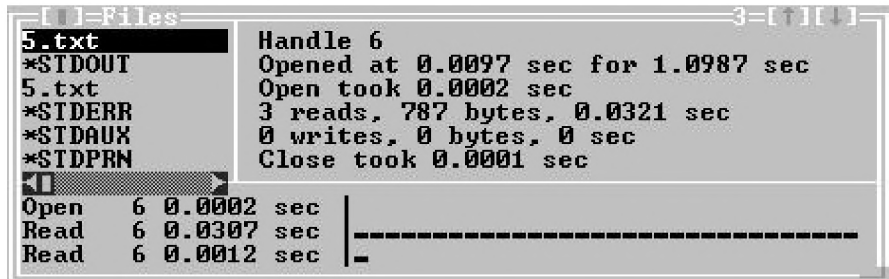
Tabelle 2.4:
Ausgabeformate der
Interrupt-Statistiken

Format	Funktion
<i>Time</i>	stellt den Zeitraum dar, der in einem Interrupt und dessen Unterrouinen zugebracht wurde.
<i>Calls</i>	zeigt an, wie oft ein Interrupt und dessen Unterrouinen aufgerufen wurden.
<i>Time and Calls</i>	zeigt sowohl Anzahl als auch Zeitdauer der Aufrufe eines Interrupts und dessen Unterrouinen an.
<i>ts</i>	liefert eine chronologische Liste der Interrupt-Aufrufe.

Files

Im Fenster *Files* zeigt Turbo Profiler alle Informationen über die Dateizugriffe an, die während der Ausführung des Programms erfolgten. Dazu muß *Statistics/Files* auf *Enabled* eingestellt sein, damit der Profiler Informationen über die Dateizugriffe erfasst, wie das Lesen, Schreiben, Öffnen oder Schließen von Dateien.

Abbildung 2.10:
Das Fenster *Files*



Das Fenster *Files* ist in drei Ausschnitte – links oben, rechts oben sowie unten – unterteilt.

Im Ausschnitt links oben werden die Dateien (unter Berücksichtigung von STDIN und STDOUT) nach Namen sortiert aufgeführt. Wenn Sie den Auswahlbalken zu der Datei bewegen, für die Sie sich interessieren, zeigt der Ausschnitt rechts oben folgende Informationen über diese Datei an:

- die Nummer des Datei-Handles;
- den Zeitpunkt, zu dem die Datei geöffnet wurde;
- wie lange die Datei geöffnet war;
- die benötigte Zeitdauer für das Öffnen der Datei;
- die Anzahl von Lese- und Schreibzugriffen in dieser Datei;
- die Gesamtzahl der gelesenen und geschriebenen Bytes;
- die gesamte Zeitdauer für alle Lese- und Schreibzugriffe;
- die benötigte Zeitdauer zum Schließen der Datei.

Im rechten oberen Ausschnitt werden nur Informationen ausgegeben. Sie können nicht mit *[Tab]* in diesen Ausschnitt wechseln.

Im unteren Ausschnitt werden die Statistiken der Dateizugriffe (für Lesen, Schreiben, Öffnen und Schließen) als einzelne Einträge und nicht als Summe für den links oben ausgewählten Datei-Eintrag dargestellt. Jeder Eintrag steht für einen bestimmten Dateizugriff.

Die beiden aktiven Ausschnitte des Fensters *Files* haben jeweils ein lokales Menü.

Collection (oberer Ausschnitt)

Mit dem Befehl *Collection* wird die Erfassung von Meßdaten über die Dateizugriffe für die aktuelle Datei (die im linken oberen Fenster hervorgehoben ist) ein- oder ausgeschaltet.

[Strg][C]

Jeder Eintrag des unteren Ausschnitts enthält Informationen über eine bestimmte Art von Datei-Operation.

Detail (oberer Ausschnitt)

[Strg][D]

Mit dem Befehl *Detail* wird festgelegt, ob die Statistiken über die Dateizugriffe detailliert aufgelistet werden oder nicht. In der ausführlichen Darstellung wird jeder Schreib- und Lesezugriff einzeln aufgeführt, wobei der Anfang des Dateizugriffs zusammen mit der Anzahl der übertragenen Bytes angegeben wird. Wenn *Detail* ausgeschaltet (*Disabled*) ist, wird nur protokolliert, wann die Datei geöffnet oder geschlossen wurde. Alle Lese- und Schreibzugriffe werden zusammengefaßt.

When Full (oberer Ausschnitt)

[Strg][W]

Der Befehl *When Full* gibt an, was passieren soll, wenn der Speicher, der für die Erfassung der Dateizugriffe vorgesehen ist, nicht ausreicht.

Wrap bedeutet, daß die am längsten zurückliegenden Dateizugriffe durch die Informationen über die letzten Dateizugriffe überschrieben werden, wenn der Speicher überläuft.

Stop bedeutet, daß die Erfassung der Dateizugriffe beendet wird, wenn der Speicher voll ist.

Display (unterer Ausschnitt)

[Strg][D]

Das lokale Menü des unteren Ausschnitts im Fenster *Files* hat nur einen Menüpunkt, mit dem das nachfolgend dargestellte Dialogfenster *Display Options* aufgerufen wird.

In diesem Dialogfenster können Sie zwei Gruppen von Schaltfeldern einstellen: *Display* und *Sort*.

- *Display* bestimmt, wie die Statistiken über die Dateizugriffe im unteren Ausschnitt dargestellt werden:
- *Graph* zeigt die *gesamte* Zeitdauer eines Zugriffs als Balkendiagramm.
- *Detail* gibt die Zugriffsdauer in Sekunden an.

Beide Optionen zeigen die Ausführungsdauer in Sekunden an; *Graph* zeigt sie jedoch graphisch an, und *Display* zeigt im Verlauf der Programmausführung an, wann Dateizugriffe erfolgen.

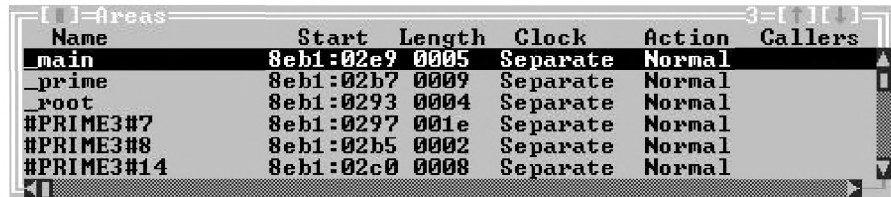
- *Sort* legt die Reihenfolge fest, in der Turbo Profiler die Statistiken ausgibt.
- *Start Time* sortiert die Dateizugriffe chronologisch in der Reihenfolge ihres Auftretens.
- *Duration* sortiert die Dateizugriffe nach der Zeitdauer, die zum Öffnen, Lesen, Schreiben oder Schließen einer Datei benötigt wurde.

Areas

Im *Areas*-Fenster stellt Turbo Profiler die markierten Bereiche ausführlich dar. Verwenden Sie dieses Fenster zur Überprüfung, ob die kurzerhand im Fenster *Module* mit den Befehlen *Add* bzw. *Remove Areas* vorgenomme-

nen Markierungen auch wirklich die gewünschten Bereiche entfernt oder hinzugefügt haben. Außerdem lassen sich in diesen Fenstern die Operationen in bestimmten Bereichen individuell einstellen.

Abbildung 2.11:
Das Fenster *Areas*



Name	Start	Length	Clock	Action	Callers
main	8eb1:02e9	0005	Separate	Normal	
prime	8eb1:02b7	0009	Separate	Normal	
root	8eb1:0293	0004	Separate	Normal	
#PRIME3#7	8eb1:0297	001e	Separate	Normal	
#PRIME3#8	8eb1:02b5	0002	Separate	Normal	
#PRIME3#14	8eb1:02c0	0008	Separate	Normal	

Standardmäßig werden im Fenster *Areas* die Bereiche in alphabetischer Reihenfolge aufgeführt. In der Praxis werden diese Bereiche normalerweise durch die entsprechenden Routinen bezeichnet. Falls jedoch jede einzelne Zeile in einer Routine markiert wurde, hat die Bezeichnung des Bereichs folgendes Aussehen:

`ModName#Dateiname#NN`

wobei *ModName* den Namen des Moduls angibt, *Dateiname* den Namen der Datei bezeichnet und *NN* die Zeilennummer darstellt. Wenn Sie eine Zeile mit einem Label versehen haben (beispielsweise mit dem Namen einer Routine), verwendet Profiler das Label als Bereichsnamen.

Das *Areas*-Fenster enthält folgende Informationen zu den markierten Bereichen:

- *Start*: sedezimal angegebene Startadresse.
- *Length*: sedezimal angegebene Länge in Bytes.
- *Clock*: Festlegung für getrennte oder kombinierte Zeiterfassung übergeordneter Routinen.
- *Action*: Maßnahme, die beim Passieren des Bereichs zu treffen ist.
- *Callers*: Festlegung für die Art der Erfassung von Aufrufinformationen.
- *Winproc*: „Yes“, wenn es sich bei dem Bereich um eine Windows-Prozedur handelt; andernfalls leer. Diese Spalte bleibt nur bestehen, wenn mit TPROFW eine Windows-Prozedur analysiert wird oder wenn TPROF als Windows Remote-Profiler agiert.

Das Fenster *Areas* bietet mehr als nur die rein statische Darstellung von Informationen. Mit dem lokalen Menü können Sie:

- Bereiche hinzufügen oder entfernen;
- Bereiche untersuchen;
- Optionen für einzelne Bereiche ändern;
- die angezeigten Bereichsinformationen sortieren.

Add Areas

Wählen Sie *Add Areas*, um Bereiche zu markieren. Dieser Befehl führt zu einem anderen Menü, das die Befehle *All Routines*, *Module* und *Routine* enthält.

[Strg][A]

- *All Routines* setzt Markierungen in jede Routine des aktuellen Moduls.
- *Module* führt zum Dialogfenster *Pick a Module*. Mit diesem Kommando können Sie Markierungen in ein anderes als das im Module-Fenster befindliche Programmmodul setzen.

(Weitere Informationen über das Dialogfenster *Pick a Module* finden Sie im Abschnitt über das Kommando *Module*).

- *Routine* führt zum Eingabefeld *Add in Enter Routine Name*. Geben Sie den Namen der Routine ein, die Sie auswählen wollen und wählen Sie dann OK.

Remove Areas

[Strg][R]

Mit dem Befehl *Remove Areas* werden Markierungen entfernt.

All Areas entfernt die Markierungen aus allen Bereichen des Programms, einschließlich der Module, die im Moment nicht im *Module*-Fenster dargestellt werden.

Module führt zum Dialogfenster *Pick a Module*. Wählen Sie in diesem Dialogfenster ein Modul, dessen Bereichsmarkierungen Sie löschen wollen.

- *This Area* entfernt die Bereichsmarkierung, die im *Areas*-Fenster gerade hervorgehoben dargestellt wird.

Inspect

[Strg][I]

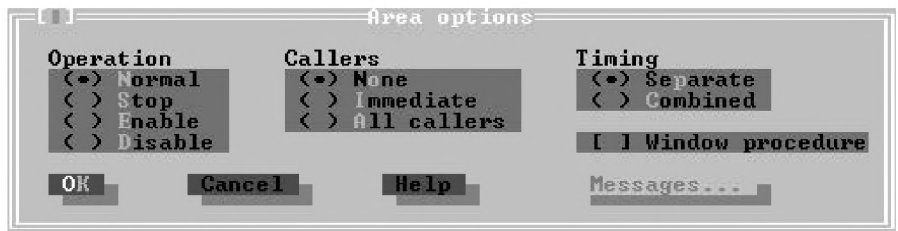
Wenn Sie *Inspect* auswählen, schaltet der Profiler in das *Module*-Fenster um und stellt den Cursor in die erste Zeile des Quellcodes mit dem entsprechenden Bereich. Wenn der hervorgehoben dargestellte Bereich keiner Programmzeile aus dem Quellcode entspricht, wird stattdessen das CPU-Fenster geöffnet.

Options

[Strg][O]

Nach Auswahl von *Options* im lokalen Menü von *Areas* erscheint das Dialogfenster *Area Options*.

Abbildung 2.12:
Das Dialogfenster
Area Options



Sie können in diesem Dialogfenster drei Gruppen von Schaltfeldern einstellen: *Operation*, *Callers* und *Timing*.

- In *Operation* legen Sie fest, welche Analysen im ausgewählten Bereich vorgenommen werden sollen.

Das Dialogfeld der Option *Operation* sehen Sie in Abbildung 2.4.

- *Callers* legt die Ausführlichkeit der Aufrufinformationen der gegenwärtigen Routine fest:
- Mit *All Callers* werden alle verfügbaren Aufrufinformationen für die aktuelle Routine festgehalten.
- *Immediate Callers* zeichnet lediglich die direkt übergeordnete Routine auf.
- *None* schaltet die Erfassung von Aufrufinformationen für die aktuelle Routine aus.
- Mit *Timing* wird festgelegt, ob der Profiler die Ausführungsdauer des aktuellen Bereichs einem übergeordneten Bereich zuordnet oder nicht.
- Mit *Windows Procedure* macht man den aktuellen Bereich als Windows-Prozedur kenntlich.
- Das Feld *Messages* wird aktiv, wenn das Markierungsfeld *Windows Procedure* aktiviert wurde. Wird *Messages* gewählt, so erscheint das Dialogfenster *Windows Procedure Messages*.

Eine vollständige Behandlung des Dialogfensters *Window Procedure Messages* finden Sie in Anhang D.

Sort

[Strg][S]

Mit Hilfe des Befehls *Sort* lassen sich die im Fenster *Areas* angezeigten Informationen nach anderen Kriterien anordnen. Die Bereiche können alphabetisch (nach *Name*) oder numerisch (nach aufsteigenden Adressen) sortiert werden. Mit *Address* stimmt die Anordnung der Bereiche eher mit der Reihenfolge im Quellcode überein.

Routines

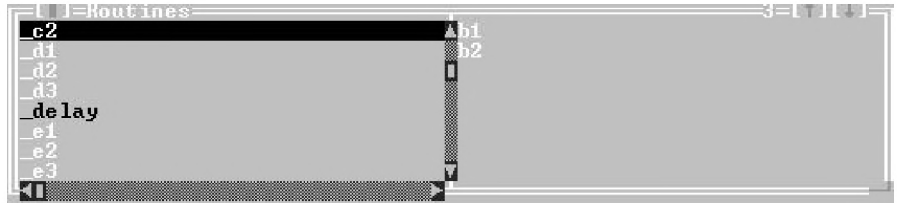
Im Fenster *Routines* stellt Turbo Profiler alle Routinen dar, die als Bereich markiert werden können. Verwenden Sie dieses Fenster, wenn Sie sich an den Namen einer Routine nicht mehr genau erinnern können oder wenn Sie überprüfen wollen, welche Routinen als Bereich markiert wurden. Mit Hilfe des Befehls *Inspect* gelangen Sie zu den Routinen der anderen Module.

Als Informationen stehen alle global verfügbaren Symbole aus der Symboltabelle der Programmdatei zur Verfügung. Diese Symbole enthalten alle Routinen- und Prozedurbezeichnungen in den Standard-Bibliotheken von Borland C++ und Turbo Pascal sowie die Namen der Routinen in den Bibliotheken anderer Hersteller (falls diese Bibliotheken symbolische Informationen enthalten).

Hinweis

Diese Menüoption erlaubt eine einfache Zuordnung zwischen den Informationen und einem Symbol.

Abbildung 2.13
Das Dialogfenster
Routines



Das Fenster *Routines* ist in zwei Ausschnitte – einen linken und einen rechten – untergliedert. Der linke Ausschnitt enthält die globalen Routinen des gesamten Programms, das untersucht wird. Der rechte Ausschnitt zeigt die Routinen, die im aktuellen Programmmodul lokal vorliegen.

Zu den lokalen Routinen gehören alle in Prozeduren verschachtelten Funktionen und Prozeduren in Pascal sowie die statisch deklarierten Funktionen in C. Globale Routinen, für die der Quellcode vorhanden ist, werden im rechten Ausschnitt hervorgehoben dargestellt. (Normalerweise steht vor allen globalen Programmvariablen in Borland C++ ein Unterstrich.)

Beide Ausschnitte des Fensters *Routines* haben lokale Menüs.

Local Module (rechter Ausschnitt)

[Strg][L] Wenn Sie den Befehl *Local Module* im rechten Ausschnitt, *Global Routines*, wählen, erscheint das Dialogfenster *Pick a Module*, in dem alle Module Ihres Programms aufgelistet sind.

Nachdem Sie die Auswahl eines Moduls mit OK abgeschlossen haben, zeigt der Profiler die lokalen Symbole des Moduls im rechten Ausschnitt des Fensters *Routines* an.

Areas (beide Ausschnitte)

[Strg][A] Der Befehl *Areas* öffnet das *Areas*-Fenster und stellt den Auswahlbalken auf die aktuelle Routine (die im Fenster *Routines* hervorgehoben war).

Callers (beide Ausschnitte)

[Strg][C] Mit dem Befehl *Callers* wird ein *Callers*-Fenster geöffnet, das alle aufrufenden Routinen der aktuellen Routine anzeigt.

Module (beide Ausschnitte)

[Strg][M] Der Befehl *Module* öffnet ein *Module*-Fenster und zeigt den Quellcode der aktuellen Routine an.

Profile (beide Ausschnitte)

[Strg][P] Das Kommando *Profile* öffnet das Fenster *Execution Profile* und zeigt die Analysestatistiken für die aktuelle Routine.

Disassembly (CPU)

Das Fenster *Disassembly* (erscheint auf dem Bildschirm mit der Bezeichnung *CPU*) zeigt den aktuellen Bereich im Fenster *Module* als disassemblierten Quellcode. In der Titelzeile des Fensters *Disassembly* wird der Prozessortyp Ihres Systems angezeigt und, falls Ihr Programm im Protected Mode läuft, auch die Bezeichnung *Protected*. Das Fenster *Disassembly (CPU)* dient dazu, Programmteile aufzuspüren, für die sich eine Umcodierung in Assembler lohnen könnte.

Abbildung 2.14:
Das Fenster
Disassembly (CPU)

```

[ ]=CPU 80486
_prime: int prime(int n)
cs:02B7 C8020000  enter    0002,00
cs:02BB 56          push     si
cs:02BC 57          push     di
cs:02BD 8B7E04      mov      di,[bp+04]
#PRIME3#14: limit = root(n);
cs:02C0 57          push     di
cs:02C1 E8CFFF      call     _root
cs:02C4 59          pop      cx
cs:02C5 8946FE      mov      [bp-02],ax
#PRIME3#15: for <i=2; i <= limit; i++)
cs:02C8 BE0200      mov      si,0002

```

Im linken Bereich der disassemblierten Zeilen steht die Adresse des Befehls, entweder sedezimal als *Segment:Offset*-Kombination oder im Falle einer Übereinstimmung mit dem aktuellen CS-Register als Adresse der Form *CS:Offset*. Wenn das Fenster groß genug (durch Vergrößern) ist, werden auch noch die Bytes dargestellt, aus denen sich der Befehl zusammensetzt. Der disassemblierte Befehl steht rechts in der Zeile.

Im Fenster *Disassembly (CPU)* werden die globalen Symbole einfach mit Ihrer Symbolbezeichnung angegeben. Statische Symbole werden allgemein in der Form

```

ModName#SymbolName      /* Borland C++ */
ModName.SymbolName      { Turbo Pascal }

```

angegeben, wobei *ModName* die Bezeichnung des Moduls und *SymbolName* die Bezeichnung für das statische Symbol ist. Die Zeilennummern haben (im allgemeinen) folgenden Aufbau:

```

ModName#Zeilennummer    /* Borland C++ */
ModName.Zeilennummer    { Turbo Pascal }

```

ModName ist wiederum die Bezeichnung des Moduls und *Zeilennummer* enthält die Nummer (dezimal) der Zeile.

Im Fenster *Disassembly (CPU)* kann man mit Hilfe der Funktionstaste [F2] einen Bereich für jeden einzelnen Maschinenbefehl markieren, der sich auf diese Weise exakt überwachen läßt. Jeder Befehl, der in dieser Weise markiert wurde und keine Symbolbezeichnung (oder ein Label) aufweist,

wird im *Areas*-Fenster als Sedezimaladresse der Form *Segment:Offset* angegeben. ([F2] kann auch dazu verwendet werden, Markierungen aus einem Bereich zu entfernen.)

Mit den lokalen Menübefehlen des Fensters *Disassembly (CPU)* können Sie sehr schnell zu den folgenden gesuchten Programmstellen springen:

- eine bestimmte Adresse, deren Code untersucht werden soll;
- die aktuelle Programmposition (CS:IP);
- die Zieladresse des aktuellen Befehls;
- die vorherige Adresse des Befehlszeigers (IP);
- die entsprechende Adresse im Quellcode.

Sie können auch eine lokale Menüoption auswählen, um das *Module*-Fenster zu aktivieren und den Cursor in der Quelldatei zum gewünschten Befehl setzen, oder die disassemblierten Befehle und den Quellcode auf drei verschiedene Arten betrachten. Der Aufruf und die Bedienung des lokalen Menüs erfolgt in der üblichen Weise.

Goto

[Strg][G]

Nach Auswahl von *Goto* erscheint ein Dialogfenster, in dem Sie nach der Adresse gefragt werden, zu der Sie springen wollen. Geben Sie eine hexadezimale Adresse im Format der Sprache ein, in der Ihr Programm verfaßt wurde.

Sie können auch Adressen eingeben, die außerhalb Ihres Programms liegen, um Code im BIOS, in DOS oder speicherresidenten Programmen zu untersuchen.

Mit dem Befehl *Previous* können Sie wieder an die alte Position im Fenster *Disassembly (CPU)* zurückkehren, an der Sie sich befanden, bevor Sie den Befehl *Goto* ausgeführt haben.

Origin

[Strg][O]

Den Befehl *MOrigin* verwenden Sie, um den Auswahlbalken im Fenster an die Programmstelle zu setzen, die durch das Registerpaar CS:IP adressiert wird. Dieser Befehl ist praktisch, wenn man sich im Code etwas umgesehen hat und wieder zur aktuellen Position (CS:IP) zurückkehren will, an der das Programm angehalten wurde.

Das Kommando *Previous* stellt das Disassembly-Fenster (das CPU-Fenster) an der Position wieder her, die es hatte, bevor Sie *Origin* gewählt haben.

Follow

[Strg][F]

Der Befehl *Follow* positioniert den Auswahlbalken des Fensters *Disassembly (CPU)* an die Zieladresse des aktuell hervorgehobenen Befehls. Der Inhalt des Fensters verschiebt sich zu der Adresse, an die die Programmsteuerung übergeben wird, wenn der hervorgehobene Befehl ausgeführt wird.

In bedingten Verzweigungen wird die Adresse angezeigt, die bei der Durchführung des Sprungs in Frage kommt.

Dieser Befehl kann für alle CALL-, JMP-, bedingte Verzweigungs- (JZ, JNE, LOOP, JCXZ) und INT-Befehle verwendet werden.

Mit Hilfe von *Previous* können Sie wieder an die Position im Fenster *Disassembly (CPU)* zurückkehren, an der Sie vor der Ausführung des Befehls *Follow* waren.

Previous

[Strg][P]

Wenn Sie einen Befehl eingeben, der die Adresse des Programmzeigers ändert (*Goto*, *Origin* oder *Follow*), können Sie mit Hilfe des Befehls *Previous* wieder an die alte Stelle zurückspringen. Wenn Sie mit Hilfe der Pfeiltasten und den Tasten [Bild oben] und [Bild unten] in dem Fenster bewegen, vergißt Turbo Profiler zwar die Position des Fensters, kann jedoch immer wieder an den Ausgangspunkt zurückspringen, da es sich dabei um die Position CS:IP handelt.

Durch wiederholte Verwendung von *Previous* können Sie zwischen zwei Punkten hin- und herspringen.

View Source

[Strg][V]

Mit dem Befehl *View Source* wird ein *Module*-Fenster geöffnet, das den Quellcode der aktuellen Routine anzeigt.

Mixed

[Strg][F]

Die disassemblierten Befehle und der Quellcode lassen sich im Fenster *Disassembly (CPU)* auf drei Arten anzeigen. Das Ausgabeformat läßt sich mit dem lokalen Menübefehl *Mixed* zwischen den drei Einstellungen *No*, *Yes* und *Both* umschalten:

- *No* bedeutet, daß kein Quellcode angezeigt wird, sondern nur der disassemblierte Code.

Im Modus *No* werden aber immer die globalen Label-Bezeichnungen anstelle der Adresse für Unterprogrammaufrufe, Verzweigungen und Datenzugriffe verwendet.

- *Yes* bedeutet, daß der Quellcode vor dem ersten disassemblierten Befehl der Zeile im Quellcode steht.

Der Profiler verwendet automatisch die Einstellung *Yes*, wenn das aktuelle Modul in einer höheren Programmiersprache verfaßt wurde.

- *Both* bedeutet, daß die Zeilen im Quellcode die disassemblierten Zeilen in den Fällen ersetzen, in denen der Quellcode vorhanden ist; sonst erscheinen nur die disassemblierten Befehle.

Der Profiler stellt automatisch auf *Both* um, wenn es sich beim aktuellen Modul um ein Assembler-Modul handelt.

Verwenden Sie *Both*, wenn Sie ein Assembler-Modul untersuchen und den ursprünglichen Quellcode anstelle der entsprechenden disassemblierten Befehle sehen wollen.

Text File

Das Fenster *Text File* dient zum Untersuchen und Editieren beliebiger ASCII-Dateien. Binär codierte Dateien zeigt dieses Fenster zwar an, stellt sie allerdings, da die Ausgabe lediglich als ASCII-Text erfolgt, als ein buntes Sammelsurium aller möglichen Zeichen des erweiterten IBM-Zeichensatzes dar.

Die Option *View/Text File* aus dem Hauptmenü öffnet zunächst ein Dialogfenster zur Dateiauswahl. Darin enthalten ist ein Eingabefeld, in welches Sie entweder mit den DOS-üblichen Jockerzeichen eine Dateimaske zum Aktualisieren der Dateiliste, oder gleich den Namen der gewünschten Datei eingeben.

Die Arbeitsweise dieses Eingabefeldes ist derjenigen des weiter oben behandelten Eingabefeldes *Program Load* sehr ähnlich.

Nachdem Sie sich für eine Datei entschieden haben, gibt Turbo Profiler diese Datei im File-Fenster aus.

Das Fenster *File* zeigt den Inhalt der von Ihnen ausgewählten Datei. Im oberen Teil des Fensters wird der Name der Datei und die Nummer der Zeile ausgegeben, in der sich der Cursor gerade befindet.

Das lokale Menü des Fensters Text File

Das lokale Menü des Fensters *Text File* beinhaltet mehrere Optionen für die Cursor-Positionierung, die Änderung der Darstellung und das Editieren der Datei.

Goto

[Strg][G]

Positioniert den Cursor auf die von Ihnen angegebene Zeile innerhalb der Datei. Wenn Sie eine Zahl angeben, die größer ist als die höchste (letzte) Zeilennummer, dann wird der Cursor auf das Dateiende positioniert.

Search

[Strg][S]

Sucht ab der aktuellen Cursor-Position bis zum Dateiende nach einer Zeichenfolge. Sie werden aufgefordert, einen Suchstring einzugeben. Wenn Sie mit Hilfe der Taste [Einfg] einen Block in der Datei markiert haben, wird dieser Block zur Initialisierung des Dialogfensters *Search* verwendet. Damit können Sie sich die Eingabe von Strings sparen, die bereits in der Datei stehen. Der Such-String kann einfache Jockerzeichen enthalten; ein ? steht für nur ein Zeichen, ein * für 0 oder mehrere Zeichen.

Die Suche beginnt bei der aktuellen Cursorposition; wenn das Dateiende erreicht ist, setzt sie sich nicht automatisch am Dateianfang fort. Soll die Suche über die gesamte Datei hinweg erfolgen, müssen Sie demzufolge

den Cursor mit **[Strg][Bild oben]** zunächst auf die erste Zeile der Datei positionieren.

Zusätzlich zur herkömmlichen Methode, läßt Turbo Profiler zur Eingabe eines Such-String auch noch folgendes zu: tippen Sie einfach die gesuchte Zeichenfolge ein. Dadurch wird ein Dialogfenster geöffnet, genau so, als ob Sie die Option *Search* aktiviert hätten.

Next

[Strg][N]

Sucht in der aktuellen Datei nach dem nächsten Vorkommen einer über die Option *Search* eingegebenen Zeichenfolge. Dieser Befehl kann daher erst aufgerufen werden, wenn zuvor bereits ein Such-Vorgang durchgeführt wurde.

Diese Funktion ist hilfreich, wenn das *Search*-Kommando nicht die gewünschte Stelle der eingegebenen Zeichenfolge fand. Sie können das Kommando so oft eingeben, bis Sie gefunden haben, wonach Sie suchen.

File

[Strg][F]

Öffnet das gleiche Dialogfenster wie auch die Option *View/Text File* und bietet Ihnen dadurch die Möglichkeit, eine andere Datei zu laden

Edit

[Strg][S]

Eröffnet Ihnen die Möglichkeit, Änderungen in der geladenen Datei vorzunehmen. Dabei wird der externe Editor, den Sie während der Installation mit TFINST in dem Feld *Editor Program Name* des Dialogfensters *Options/ Directories* eingegeben haben, aufgerufen.

Diese Option steht nicht zur Verfügung, wenn mit TPROFW Windows-Anwendungen analysiert werden.

Coverage-Analyse

Das Fenster *Coverage* besteht aus zwei Abschnitten: Der linke Abschnitt enthält eine Liste ausgewählter Module, der rechte zeigt die Blöcke, die in diesen Modulen enthalten sind. Ein *Block* ist ein Code-Abschnitt, der nur einen einzigen Einsprungpunkt und einen Ausgang hat; es gibt keine Sprünge in einen Block hinein oder aus einem Block heraus.

Abbildung 2.15:
Das Fenster
Coverage



Durch die Auswahl des Modus *Coverage* (*Statistics/Profiling Options*) wird das Fenster *Coverage* geöffnet; es ersetzt das Fenster *Execution Profile*. Standardmäßig befinden sich alle Programmmodule im linken Abschnitt und die in diesen Modulen enthaltenen Blöcke im rechten.

Hinweis

Das Fenster *Coverage* und das Fenster *Execution Profile* schließen sich wechselseitig aus; es kann nur eines von beiden geöffnet sein.

Per Voreinstellung enthält das *Coverage*-Fenster nur die nicht ausgeführten Blöcke (im rechten Ausschnitt). Blöcke, die während eines Programmdurchlaufs ausgeführt wurden, werden aus der Liste entfernt.

Bevor ein Programm läuft, werden alle Programmblöcke im *Module*-Fenster mit einem Pfeil markiert. Im Standardmodus entfernt Turbo Profiler den Pfeil, sobald ein Modul ausgeführt worden ist. Sie können also dafür sorgen, daß die markierten Blöcke im Rahmen mehrerer Durchläufe der Reihe nach verschwinden.

Hinweis

Turbo Profiler markiert automatisch alle Programmblöcke mit einer einfachen Markierung.

Wenn Sie im Dialogfenster *Profiling Options* (Sie erreichen es über das lokale Menü von *Statistics*) einen Coverage-Zähler eingerichtet haben, wird die pfeilförmige Markierung erst entfernt, nachdem der Block die vorgegebene Anzahl an erfolgreichen Aufrufen hinter sich hat.

Um alle Blöcke des Programms in den Zustand „Noch nie aufgerufen“ zu versetzen, wählen Sie im *Statistic*-Menü den Punkt *Delete All*.

Jeder Abschnitt des Coverage-Fensters hat ein eigenes lokales Menü.

Das lokale Menü des linken Abschnitts enthält Kommandos, mit denen Sie vorgeben können, welche Module in eine Analyse aufgenommen werden sollen. Das lokale Menü des rechten Abschnitts enthält Kommandos, mit denen die Darstellung der Blöcke gesteuert wird.

Add All Modules (linker Abschnitt)

[Strg][A]

Wenn Sie *Add All Modules* wählen, werden die Blöcke aller Module des ausführbaren Programms für die Analyse markiert. *Add All Modules* ist die Voreinstellung, wenn aus dem Dialogfenster *Profiling Options* heraus der *Coverage*-Modus gewählt wurde.

Remove All Modules (linker Abschnitt)

[Strg][R]

Remove All Modules entfernt alle Module aus der Modulliste; anschließend sind beide Hälften des *Coverage*-Fensters leer.

Add Module (linker Abschnitt)

[Strg][M]

Das Kommando *Add Module* öffnet das Dialogfenster *Pick a Module*; es erscheint eine Liste mit den Modulen, die zu diesem Programm gehören; Sie können nun das gewünschte Modul auswählen. Weitere Informationen über das Dialogfenster *Pick a Module* und dessen Verwendung finden Sie im entsprechenden Abschnitt.

Remove Module (linker Abschnitt)
 [Strg][V] Das Kommando *Remove Module* öffnet das Dialogfenster *Pick a Module*. Mit Hilfe dieses Dialogfensters können Sie ein Modul von weiteren Analysedurchläufen ausschließen.

Delete This Item (linker Abschnitt)
 [Strg][D] Das Kommando *Delete This Item* entfernt das im Moment hervorgehoben dargestellte Modul aus den noch stattfindenden Analysedurchläufen. Wenn dieses Kommando ausgeführt wird, werden alle Blöcke des entfernten Moduls aus dem rechten Abschnitt des *Coverage*-Fensters entfernt.

Display (rechter Abschnitt)
 [Strg][D] Mit dem Kommando *Display* (rechter Abschnitt des lokalen Menüs) holen Sie das Dialogfenster *Coverage Display* auf den Bildschirm.

Abbildung 2.16:
Das Dialogfenster
Coverage



- Die Display-Aktivierungsschalter definieren, was im rechten Abschnitt des Coverage-Fensters erscheinen soll:
 - Wenn Sie *All* wählen, werden alle Blöcke (aufgerufene und nicht aufgerufene) im rechten Abschnitt aufgeführt. Bei den Blöcken, die aufgerufen wurden, wird angegeben, wie oft sie aufgerufen wurden; Blöcke, die nicht aufgerufen wurden, sind mit einer Null gekennzeichnet. Sie können im Dialogfenster *Profiling Options* (Option *Maximum Coverage Count*) einen Zähler setzen, der von Turbo Profiler ständig verglichen wird. In Abbildung 2.18 finden Sie weitere Informationen zu der Einstellung der Zählrate von Turbo Profiler.
 - Die Standardeinstellung lautet *Not Hit* und bedeutet, daß nur Blöcke aufgeführt werden, die nicht aufgerufen wurden.
- Die *Sort*-Schalter legen die Reihenfolge fest, nach der die Blöcke geordnet werden. Diese Wahlmöglichkeit steht nur zur Verfügung, wenn *Display* auf *All* geschaltet wurde und wenn in dem Dialogfenster *Profiling Options* der Wert *Max Coverage Count* größer als 1 ist.
 - *Address* bedeutet, daß die Blöcke alphabetisch geordnet werden.
 - *Count* bedeutet, daß die Blöcke nach der Häufigkeit ihrer Aufrufe geordnet werden; nicht aufgerufene Blöcke erscheinen zuerst.
- Im Schaltfeld *Group* können Sie die Blöcke einzeln, nach ihrer Zugehörigkeit zu Routinen oder zu Modulen ordnen

Hinweis

Das Schaltfeld *Group* steht nur dann zur Verfügung, wenn *Display* auf *All* eingestellt ist.

- Bei der Einstellung *Block* (der Standardeinstellung) erscheinen alle Blöcke im Fensterausschnitt.
 - Bei der Einstellung *Routine* werden die Blöcke ihrer Routinenzugehörigkeit nach gruppiert. In diesem Fall erscheinen nur Routinenamen aus den Modulen, die auch im rechten Abschnitt aufgeführt werden. Außer den Routinenamen erscheint allerdings auch noch die Anzahl der in der jeweiligen Routine enthaltenen Blöcke und die Gesamtzahl der aufgerufenen Routinen.

Abbildung 2.17:
Das Fenster
Coverage,
Gruppierung der
Blöcke nach
Routinen

[] = Coverage				
All	root	Hit	2 of 3	-----
	prime	Hit	8 of 8	=====
	main	Hit	7 of 7	=====

- Der Aktionsschalter *Module* funktioniert ganz ähnlich wie der Aktionsschalter *Routine*: es werden alle Blöcke in Gruppen zusammengefaßt. Allerdings werden beim Schalter *Module* die Blöcke ihrer Modulzugehörigkeit nach gruppiert und nicht entsprechend ihrer Routinenzugehörigkeit.

Position (rechter Abschnitt)

[Strg][P]

Das Kommando *Position* wird verwendet, um den Quellcode eines bestimmten Code-Blocks zu betrachten. Gehen Sie in den rechten Abschnitt und wählen Sie einen Code-Block aus (hell hinterlegen!). Anschließend wählen Sie *Position*. Daraufhin erscheint der Quellcode im *Module*-Fenster. Das *Module*-Fenster wird durch dieses Kommando jedoch nicht aktiviert.

Das Kommando *Position* wird auch durch die Leertaste ausgelöst.

Module (rechter Abschnitt)

[Strg][M]

Das Kommando *Module* funktioniert ganz ähnlich wie das Kommando *Position*, aktiviert jedoch das *Module*-Fenster und positioniert den Cursor an den Anfang des ausgewählten Blocks.

Dieser Befehl kann auch durch Drücken der Taste [Eingabe] aktiviert werden.

Hinweis

Wenn der *Coverage*-Modus aktiv ist, fallen einige Funktionen von Turbo Profiler weg. Insbesondere stehen folgende Fenster im *View*-Menü nicht zur Verfügung: *Execution Profile*, *Callers*, *Files*, *Areas* und *Routines*. Das *Statistics*-Menü büßt die *Caller*- und die *Files*-Funktionen ein. Auch der *Accumulation*-Schalter im *Statistics*-Menü wird automatisch auf *Enabled* gesetzt. Außerdem werden im lokalen Menü des *Module*-Fensters die Funktionen *Add Areas*, *Remove Areas*, *Operation* und *Callers* ausgeschaltet.

Run-Menü

Im Menü *Run* gibt es drei Befehle, die sich auf den Ablauf des zu untersuchenden Programms beziehen: *Run*, *Program Reset* und *Arguments*. Der Profiler erhält die Programmsteuerung nach folgenden Ereignissen wieder:

- Das Programm wird beendet.
- Das Programm kommt an eine Bereichsmarkierung, die mit der Operation *Stop* gekennzeichnet ist.
- Das Programm wird von Ihnen durch die Interrupt-Taste beendet.
- Ihr Programm erzeugt eine Ausnahmebedingung (*Exception*).

(Normalerweise wird als Interrupt-Taste [*Strg*][*Untbr*] verwendet. Mit TFINST, dem Installationsprogramm von Turbo Profiler, können Sie eine andere Tastenkombination auswählen. Wenn Sie mit Hilfe von TPROFW eine Windows-Anwendung analysieren, unterbrechen Sie das Programm mit [*Strg*][*Alt*][*S-Abf*]. (Näheres hierzu finden Sie in Anhang D.)

Sie können Ihr Programm auch dann starten, wenn das *Module*-Fenster nicht geöffnet ist (dazu muß jedoch ein Programm in Turbo Profiler geladen sein).

Hinweis

Wenn Sie *Run/Run* oder *Run/Program Reset* auswählen, werden alle Statistiken der vorherigen Analysen gelöscht. Zum Speichern von bestimmten Meßdaten wählen Sie den Befehl *Statistics/Save*, bevor Sie *Run* oder *Program Reset* wählen, oder verwenden Sie *Statistics/Profiling Options*, um die Anzahl der Programmdurchläufe auf größer als 1 einzustellen.

Run

Mit dem Befehl *Run* wird das Programm gestartet und dessen Laufzeitverhalten gemessen.

Falls Sie *Display Swapping* auf *Always* gesetzt haben, wird die Benutzerumgebung des Turbo Profilers durch die Ausgaben des Programms ersetzt, bis das Programm beendet ist oder Sie es unterbrochen haben.

Program Reset

Mit dem Befehl *Run/Program Reset* wird das Programm erneut vom Laufwerk geladen. Verwenden Sie diesen Befehl, wenn die Ausführung des Programms während einer Analyse zu weit gelaufen ist und noch einmal neu gestartet werden soll.

Wenn Sie *Run/Program Reset* auswählen, während das Fenster *Module* oder *Disassembly (CPU)* aktiv ist, kehrt die Anzeige in diesem Fenster nicht an den Anfang des Programm zurück. Stattdessen bleibt der Cursor an derselben Stelle, an der er sich befand, als Sie den Befehl *Program Reset* auswählten.

Arguments

Für das untersuchte Programm sind möglicherweise einige Kommandozeilenargumente erforderlich. Mit dem Befehl *Run/Arguments* können Sie

die benötigten Kommandozeilenargumente in Turbo Profiler speichern. Wenn Sie anschließend den Befehl *Run/Run* oder *Run/Program Reset* auswählen, übergibt der Profiler die Argumente an Ihr Programm, wie bei der Eingabe in der Kommandozeile. Sie können diese Argumente in Turbo Profiler ändern und das Programm neu starten.

Geben Sie die Argumente genauso wie auf DOS-Ebene ein. (Geben Sie nicht den Namen des Programms mit ein.)

Das Menü Statistics

Das Menü *Statistics* enthält Befehle zum

- Festlegen der Daten, die erfaßt werden sollen (Aufrufhierarchie, Interrupts, Overlays);
- Einstellen des Analysemodus (aktiv oder passiv oder *Coverage*);
- Bestimmen der Anzahl der Programmläufe und Bereiche;
- Ein-/Ausschalten der automatischen Datenerfassung;
- Löschen der Profiler-Statistiken;
- Speichern der Profiler-Statistiken in Dateien;
- Laden von zuvor gespeicherten Statistiken.

Die Standarderweiterung der Dateien lautet *.TFS*, Sie können jedoch jede andere Erweiterung verwenden.

Sie können alle statistischen Informationen der aktuellen Analyse in einer *.TFS*-Datei speichern. Diese *.TFS*-Datei können Sie jederzeit wieder laden, um die Ergebnisse zu untersuchen – alle Meßdaten sind wieder verfügbar, ohne daß Sie erneut eine Analyse vornehmen müssen.

Diese Eigenschaft ist vor allem dann sehr vorteilhaft, wenn die Untersuchung Ihrer Programme sehr viel Zeit beansprucht. Sie können so auch mehrere verschiedene Versionen von Analysen unter unterschiedlichen Bedingungen speichern und die einzelnen Ergebnisse später schnell vergleichen. Es ist zum Beispiel auch möglich, ein Makro zu schreiben, das automatisch fünf Analysen mit verschiedenen Optionen oder Bereichsmarkierungen durchführt, und die Ergebnisse in fünf verschiedenen *.TFS*-Dateien speichert. Das Makro kann über Nacht gestartet werden, und die Ergebnisse lassen sich am nächsten Morgen vergleichen.

Hinweis

Die ersten vier Einträge im *Statistics*-Menü sind einfache Schalter. Die Optionen *Callers*, *Files*, *Interrupts* und *Overlay* stehen auf *Enabled* oder *Disabled*.

Callers

Wenn Sie die Option *Callers* aktivieren, stellt der Profiler Statistiken darüber zusammen, von welchen Routinen aus eine Routine aufgerufen wird. Um festzulegen, welche Routinen beobachtet werden sollen, wählen Sie den Befehl *Callers* im lokalen Menü des Fensters *Module* oder den

Befehl *Options* im Fenster *Areas* aus und stellen dann das entsprechende Schaltfeld in *Callers* bzw. *Areas* ein.

Hinweis

Sie müssen Ihr Programm einmal starten und einige Statistiken erfaßt haben, damit dieses Fenster irgendwelche Informationen anzeigen kann.

Nach dem Start Ihres Programms und der Erfassung der Daten gehen Sie in das Fenster *Callers*, um sich dort die Statistik der Aufrufhierarchien anzuschauen.

Die Erfassung der Aufrufinformationen kostet viel Speicherplatz und die Programmausführung wird verlangsamt. Falls Sie keine Informationen über die Aufrufe benötigen, stellen Sie *Callers* auf *Disabled* ein.

Files

Wenn Sie die Option *Files* einschalten, erfaßt der Profiler Statistiken darüber, welche Dateien in Ihrem Programm geöffnet werden und welche Schreib- und Lese-Operationen stattfinden.

Nachdem Sie Ihr Programm gestartet haben und die Ergebnisse der Analyse vorliegen, verwenden Sie das Fenster *Files*, um die Dateizugriffe in Ihrem Programm zu überprüfen.

Informationen über die Dateizugriffe verbrauchen viel Speicherplatz. Außerdem verlangsamt sich durch deren Erfassung die Programmausführung. Wenn Sie keine Informationen über die Dateizugriffe brauchen, stellen Sie *Files* auf *Disabled* ein.

Interrupts

Wenn Sie die Option *Interrupts* einschalten, hält der Profiler fest, welche Interrupts von Ihrem Programm aufgerufen werden. Der Profiler verwaltet eine eigene Statistik für die DOS-, Video-, Disk-, BIOS- und Tastatur-Interrupts.

Nach dem Start Ihres Programms und der Erfassung der Meßdaten verwenden Sie das Fenster *Interrupts*, um nachzusehen, welche Interrupts von Ihrem Programm aufgerufen wurden.

Die Erfassung der Interrupt-Informationen belegt viel Speicherplatz und verzögert den Programmablauf. Wenn Sie also an den Interrupt-Aufrufen nicht interessiert sind, stellen Sie *Interrupts* auf *Disabled* ein.

Overlays

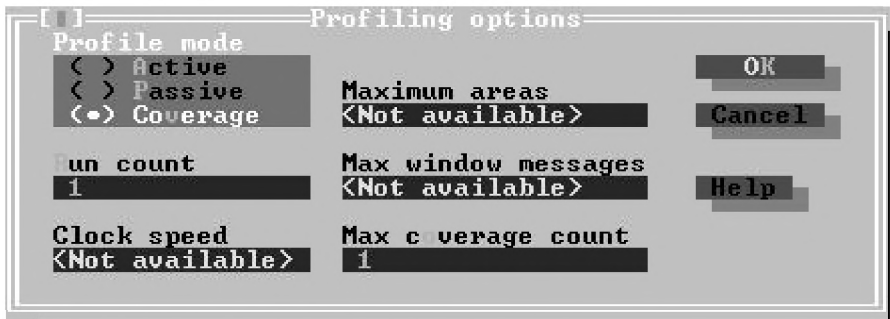
Die Option *Overlays* bestimmt, ob über das Laden von Overlays Daten erfaßt werden. Wenn Ihr Programm keine Overlays enthält, erhalten Sie eine Fehlermeldung und die Option bleibt ausgeschaltet.

Informationen zu Overlays belegen viel Platz im Speicher und die Erfassung verlangsamt die Programmausführung. Wenn Sie keine Informationen über die Overlay-Zugriffe benötigen, setzen Sie *Overlays* am besten auf *Disabled*.

Profiling-Optionen

Abbildung 2.18:
Das Dialogfenster
Profiling Options

Mit dem Befehl *Statistics/Profiling Options* wird das Dialogfenster *Profiling Options* geöffnet.



Im Dialogfenster *Profiling Options* können Sie folgende Optionen einstellen:

- *Profile Mode* legt fest, welche Art von Analyse vom Profiler durchgeführt wird. Die Voreinstellung ist *Active*.
 - *Active Analysis* bedeutet, daß der Profiler alle statistischen Informationen für eine markierte Routine erfaßt: die zeitlichen Informationen, die Anzahl der Aufrufe und von wo aus die Routine aufgerufen wurde.
 - Bei *Passive Analysis* wird nur die Zeitdauer für die markierten Routinen erfaßt; diese Art der Analyse ist unter TPROFW im erweiterten 386-Modus nicht möglich.
 - *Coverage Analysis* bedeutet standardmäßig, daß Turbo Profiler nur Code-Abschnitte protokolliert, die während der entsprechenden Programmläufe nie aufgerufen wurden.

Hinweis

Wenn Sie den Analyse-Modus von *Active* auf *Passive* umstellen (oder umgekehrt), erhalten Sie die Aufforderung „Reload Program and change profiling mode?“ Wenn Sie daraufhin den Aktivierungsschalter *Yes* betätigen, wird der bisher erzeugte Bestand an statistischen Daten gelöscht und der neue Analyse-Modus tritt in Kraft. Sichern Sie also Ihre statistischen Daten, bevor Sie den Analysemodus wechseln, oder die Daten gehen verloren.

- In *Run Count* wird festgelegt, wie oft das Programm für eine Analyse ablaufen soll. Der Vorgabewert ist 1.

Hinweis

Wenn *Run Count* auf einen Wert höher als 1 gesetzt wurde, dann führt Turbo Profiler die in diesem Wert genannte Anzahl an Durchläufen durch; dabei werden alle statistischen Daten in eine einzige .TFS-Datei geschrieben.

- *Clock Speed* enthält die Frequenz in Zeittakten pro Sekunde. Der Standardwert ist 100 Zeittakte pro Sekunde. Die Frequenz kann nur für passive Analysen eingestellt werden.
- In *Maximum Areas* steht die maximale Anzahl an Bereichen, in die ein Programm unterteilt werden kann. Der Standardwert ist doppelt so groß wie die Anzahl an Routinen im untersuchten Programm. (Wenn Sie diesen Wert erhöhen, verringert sich der Speicherbereich, der für Ihr Programm zur Verfügung steht.)
- In *Maximum Windows Message* wird der Höchstwert für die Anzahl der zulässigen Windows-Meldungen gesetzt. Standardmäßig ist hier der Wert 20 gesetzt. (Wenn Sie diesen Wert erhöhen, reduziert sich dadurch Speicherplatz für das zu untersuchende Programm.)
- In *Maximum Coverage Count* wird angegeben, wie viele Aufrufe eines Codeblocks von Turbo Profiler maximal protokolliert werden sollen. Standardmäßig gilt der Wert 1 (ein Block wurde entweder durchlaufen oder nicht). Wenn Sie diesen Wert erhöhen, verlängert sich dadurch die Laufzeit des Programms.

Mit den Optionen in diesem Dialogfenster können Sie den Ablauf der Analyse genau auf Ihre persönlichen Anforderungen einstellen.

Der aktive Modus bietet die umfassendste Analyse Ihres Programms. Dies hat jedoch den Nachteil, daß die Ausführung des Programms verlangsamt wird. Dagegen können Sie Ihr Programm im passiven Modus fast bei voller Geschwindigkeit ablaufen lassen. Sie erhalten jedoch keine Informationen mehr, wie oft oder von wo aus eine Routine aufgerufen wurde.

Wenn während der Analyse nicht genug Zeittakte erfaßt werden, kann es sein, daß die ermittelten Daten die Zeitdauer in den einzelnen Programabschnitten nicht exakt widerspiegeln. Die Genauigkeit läßt sich durch mehrere Programmausführungen verbessern, wodurch sich die Anzahl der erfaßten Daten erhöht. Des weiteren haben Sie die Möglichkeit, eine höhere Taktfrequenz zu wählen, um mehr Meßdaten zu erhalten. Dadurch erhöht sich die Genauigkeit der Zeitangaben für die einzelnen Bereiche. Das Verfahren hat aber den Nachteil, daß die Programmausführung verlangsamt wird.

Hinweis

Der Profiler mißt eigentlich nicht die Zeitdauer in den einzelnen Bereichen, sondern verwendet den Interrupt-Timer zum Zählen der Zeittakte. Nach Abschluß des Programms wandelt der Profiler die Anzahl der Zeittakte anhand des Werts in der Option *Clock Speed* in Ausführungszeiten um.

Accumulation

Die Option *Statistics/Accumulation* schaltet die automatische Datenerfassung ein oder aus. Das bedeutet, daß Sie zum einen Daten für einen Teil

der markierten Bereiche Meßdaten erfassen können, ohne daß die entsprechenden Bereichsmarkierungen entfernt werden müssen, und zum anderen die Datenerfassung nach dem Start Ihres Programm manuell einschalten können.

Um nur einen Teil der markierten Bereiche zu untersuchen, gehen Sie am besten wie folgt vor:

1. Im lokalen Menü des Fensters *Areas* wählen Sie den Befehl *Options* zum Öffnen des Dialogfensters *Area Options* aus.
2. Für die gewünschten Bereiche stellen Sie die Bereichsmarkierung von *Normal* auf *Enable* (zum Starten der Erfassung) oder auf *Disable* (zum Stoppen der Erfassung) um.
3. *Statistics/Accumulation* schalten Sie mit *Disabled* ab.
4. Starten Sie Ihr Programm. Der Profiler wird erst dann mit der Erfassung der Daten beginnen, wenn eine Bereichsmarkierung überquert wird, die auf *Enable* eingestellt ist.

Um die Datenerfassung nach dem Start Ihres Programms manuell einzuschalten, folgen Sie nachstehendem Schema:

1. Setzen Sie Bereichsmarkierungen.
2. Stellen Sie *Statistics/Accumulation* auf *Disabled* ein.
3. Starten Sie Ihr Programm vom Profiler aus (indem Sie [F9] drücken).
4. Wenn sich das Programm im gewünschten Abschnitt befindet, unterbrechen Sie das Programm (mit der Interrupt-Taste).
5. Schalten Sie dann die Erfassung der Meßdaten wieder ein (*Statistics/Accumulation* auf *Enabled* setzen).
6. Setzen Sie die Programmausführung mit [F9] fort.

Turbo Profiler beginnt sofort wieder mit der Erfassung von Meßdaten.

Wann empfiehlt es sich, die Erfassung abzuschalten?

Wenn eine Routine oder eine Routinensammlung von mehreren verschiedenen Programmstellen aus aufgerufen wird, möchten Sie vielleicht erfahren,

- wieviel Zeit eine Routine braucht,
- welcher Teil Ihres Programms die Routine aufruft,
- wieviel Zeit in einer Routine verbracht wird, nachdem ein bestimmtes Ereignis auftrat.

Um die Erfassung auf bestimmte Aufrufe zu beschränken, verwenden Sie den Befehl *Statistics/Accumulation*, um die Datenerfassung am Anfang des Programms abzuschalten. Markieren Sie einen Bereich zur Datenerfas-

sung kurz vor dem Aufruf der fraglichen Routine. Setzen Sie dazu die Bereichsmarkierung auf *Enable*. Markieren Sie einen weiteren Bereich nach der Rückkehr der Routine, um die Erfassung wieder abzuschalten. Sie können die *Enable* und *Disable*-Bereiche auch in völlig unterschiedlichen Programmteilen verwenden, und so die Erfassung nur beim Eintritt bestimmter Ereignisse aktivieren.

Beispiel 1: Die Untersuchung eines bestimmten Aufrufs

Nehmen wir einmal an, daß Sie sich bei den Aufrufen der Routine **abc** ausschließlich für die Aufrufe aus **xyz** interessieren:

```
=>  main()          /* normale Bereichsmarkierung */
    {
        :
        abc();       /* keine Datenerfassung */
        :
        xyz();
    }

=>  xyz()           /* normale Bereichsmarkierung */
    {
        :
e>   abc();         /* Erfassung einschalten */
d>   :
    }

=>  abc()           /* normale Bereichsmarkierung */
    {
        :
    }
```

Beachten Sie, daß ab e> die Erfassung eingeschaltet ist und ab d> wieder abgeschaltet ist. Vor dem Programmstart müssen Sie *Statistics/Accumulation* ausschalten, sonst registriert der Profiler versehentlich auch den ersten Aufruf von **abc** in **main**.

Beispiel 2: Die Untersuchung nach einem bestimmten Ereignis

Betrachten wir noch einen weiteren Fall: Falls sich die Routine **xyz** aufgrund einer globalen Information, die über die Routinen **bufferon** und **bufferoff** gesteuert wird, unterschiedlich verhält und nur eine Variante des Programmablaufs für uns interessant ist – wenn nämlich **bufferflag** den Wert 1 hat – könnte man wie folgt vorgehen:

```
=>  main()          /* normale Bereichsmarkierung */
    {
        :
        xyz();       /* keine Statistik erfassen */
        :
        bufferon();
    }
```

```

:
xyz();      /* Statistik dieses Aufrufs erfassen */
:
bufferoff();
:
xyz();      /* keine Statistik erfassen */
}

=>  bufferon()    /* normale Bereichsmarkierung */
{
:
    bufferflag = 1;
e>  }

d>  bufferoff()   /* normale Bereichsmarkierung */
{
:
    bufferflag = 0;
}

=>  xyz()          /* normale Bereichsmarkierung */
{
:
}

```

Beachten Sie bitte daß sich die Markierungen zum Einschalten (e>) und zum Abschalten der Erfassung (d>) nicht in der Nähe der Aufrufe von **xyz** befinden. Sie müssen natürlich die Erfassung vor dem Programmstart (mit *Disable* in *Statistics/Accumulation*) ausschalten, sonst fließt der erste Aufruf versehentlich in die Statistik ein.

Delete All

Der Befehl *Statistics/Delete All* löscht alle Statistiken, die für die aktuelle Untersuchung erfaßt wurden. Der Befehl dient in erster Linie der Bereinigung der Daten, um neu anfangen zu können. Mit *Delete All* werden alle Daten in den Fenstern (*Execution Profile*, *Callers*, *Interrupts*, *Files* und *Overlays*), aber nicht die Einstellungen für die Analyse gelöscht.

Save

Mit dem Befehl *Statistics / Save* werden folgende Daten, Einstellungen und Optionen gespeichert:

- alle Statistiken, deren Erfassung während der aktuellen Analyse aktiviert war (Zeitdauer und Anzahl der Ausführung, Aufrufinformationen, Dateizugriffe, Interrupts und Overlays);
- alle Bereichsinformationen (Bereichsname, Operationen, Aufrufoptionen, getrennte oder kombinierte Zeiterfassung), die im Fenster *Execution Profile* angezeigt werden.

Sobald Sie die Statistiken in einer Datei gespeichert haben, können Sie die Daten jederzeit mit dem Befehl *Statistics/Restore* wieder einlesen.

Wenn Sie *Statistics/Save* auswählen, erscheint das Dialogfenster *Enter File Name To Save*.

Im Eingabefeld *Name* ist der Standard-Name für die .TFS-Datei vorgegeben (*ProgName.TFS*, wobei *ProgName* die Bezeichnung des gerade untersuchten Programms ist).

Dateien speichern

Um die aktuelle Profiler-Statistik in der vorgegebenen Datei zu speichern, wählen Sie *OK* aus.

Wenn Sie diese Daten in einer anderen Datei speichern wollen, gehen Sie wie folgt vor:

1. Aktivieren Sie das Eingabefeld *File/Name*.
2. Geben Sie einen anderen Dateinamen ein (optional unter Angabe von Laufwerk und Pfad).
3. Wählen Sie *OK* (oder drücken Sie [*Eingabe*]).

Wenn die .TFS-Datei bereits vorhanden ist, erscheint eine Anfrage, ob die Datei überschrieben werden soll.

Restore

Nach Auswahl des Befehls *Statistics/Restore* erscheint das Dialogfenster *Restore* auf.

Das Dialogfenster *Restore* funktioniert wie alle anderen Dialogfenster von Turbo Profiler, mit denen Dateien geladen werden können:

- Geben Sie einen Dateinamen oder eine Dateiauswahl (mit DOS-Jokerzeichen) im Eingabefeld *File Name* an.
- Wählen Sie gegebenenfalls ein anderes Laufwerk oder einen anderen Pfad im Verzeichnisbaum aus.
- Selektieren Sie einen Dateinamen in der Dateiliste.
- Schließen Sie den Vorgang mit *OK* ab (oder verlassen Sie das Dialogfenster, ohne eine Datei zu laden, indem Sie *Cancel* auswählen).
- Mit *Help* können Sie ein Fenster mit Hilfestellungen zur Benutzung dieses Dialogfenster öffnen.

Nachdem Sie den Namen einer .TFS-Datei eingegeben oder ausgewählt haben und die Datei geladen wurde, übernimmt Turbo Profiler alle gespeicherten Daten, Statistiken und Einstellungen, die in dieser Datei gespeichert sind.

Print-Menü

Mit Hilfe des Menüs *Print* können Sie den Inhalt eines beliebigen geöffneten Fensters in eine neue oder bereits bestehende Datei oder direkt auf den Drucker ausgeben.

Statistics

Mit dem Befehl *Print/Statistics* ist es möglich, den Inhalt aller geöffneten Profiler-Fenster (mit Ausnahme von *Module*) auf einem Drucker auszugeben oder in eine Datei zu schreiben, die im Dialogfenster *Printing/Options* angegeben werden können.

Bevor Sie *Print/Statistics* auswählen, sollten Sie das Dialogfenster *Printing Options* (mit dem Befehl *Print/Options*) öffnen und die aktuellen Druckeinstellungen überprüfen (Papierformat, Ausgabeziel, Zeichensatz und den Dateinamen – falls Sie die Ausgabe in einer Datei speichern wollen).

Wenn Sie die Statistiken in eine bereits vorhandene Datei schreiben wollen, erscheint ein Menü, in dem Sie entscheiden können, ob die Ausgabe an die Datei angehängt, die Datei durch die Ausgabe überschrieben oder der Druckvorgang abgebrochen werden soll.

Module

Im Dialogfenster *Pick a Module*, das Sie über den Befehl *Print/Module* erreichen, geben Sie an, welches Ihrer Module auf dem Drucker ausgegeben oder in die Datei geschrieben werden soll, die im Dialogfenster *Printing Options* angegeben wurde.

Sie können den Namen eines bestimmten Moduls auswählen oder *All Modules* verwenden, um den gesamten verfügbaren Quellcode Ihres Programms auszugeben. Beim Ausdrucken eines Moduls wird ein *kommentiertes Listing* erzeugt, in dem die Ausführungszeit und die Anzahl der Ausführungen neben den Programmzeilen angegeben werden, die als Bereich markiert sind, wie Sie nachstehender Abbildung entnehmen können:

```

Program: C:\PROFILEPRIME1.EXE  File PRIME1.C

Time   Counts
      #include <stdio.h>

0.0090 999   prime(int n)
           {
               int i;

0.0117 999           for (i=2; i<n; i++)
1.1456 78022           if (n % i == 0)
0.0080 831               return 0;
0.0017 168           return 1;
0.0101 999       }

0.0000 1      main()
           {
               int i, n;

0.0000 1           n=1000;
0.0000 1           for (i=2; i<n; i++)
0.0255 999               if (prime(i))
4.1670 168                   printf("%d\n", i);
0.0000 1       }
```

Options

Wenn Sie im Menü *Print* den Befehl *Options* auswählen, erscheint das Dialogfenster *Printing Options*.

- *Width* ist die Anzahl an Zeichen pro Zeile (Standard = 80).
- *Height* ist die Gesamtzahl der Zeilen pro Seite (Standard = 66).
- Mit den Schaltfeldern *Printer/File* kann ausgewählt werden, ob die Ausgabe der Meßdaten an einen angeschlossenen Drucker oder in eine Datei erfolgen soll. Die Vorgabe ist *Printer*.
- Mit den Schaltfeldern *Graphics/ASCII* schalten Sie zwischen dem erweiterten IBM-ASCII-Zeichensatz (mit den Grafikzeichen) und dem Standard-ASCII-Zeichensatz um. Die Voreinstellung ist *ASCII*.
- In *Destination File* kann das Laufwerk (optional), der Pfad (optional) und der Name (muß unbedingt angegeben werden) der Druckdatei festgelegt werden.

Options-Menü

Mit den Befehlen des Menüs *Options* können Sie:

- Makros aufzeichnen;
- eines oder alle Makros löschen;
- Darstellungsoptionen ändern, die das Verhalten und das gesamte Erscheinungsbild von Turbo Profiler beeinflussen;
- zusätzliche Verzeichnisse angeben, in denen Turbo Profiler nach Quelldateien suchen soll;
- die Anordnung der Fenster, die Makros, Einstellungen in Menüs und verschiedene andere Optionen in einer Konfigurationsdatei speichern;
- Einstellungen und Optionen aus einer Konfigurationsdatei laden.

Makros

Der Befehl *Options/Macros* ruft ein weiteres Menü auf. Mit den Befehlen des Makromenüs können Sie Tastaturmakros definieren oder löschen.

Mit den Makromöglichkeiten des Profilers können Sie häufig benötigte Folgen von Tastaturbefehlen aufnehmen. Während einer Analyse werden Sie zum Beispiel oft dieselbe Befehlsfolge verwenden, um an eine bestimmte Programmstelle zu gelangen. Sie können aber ein Makro definieren, das alle Tasten vom Aufruf des Profilers ab aufzeichnet, bis sich das Programm im gewünschten Zustand befindet. Dann können Sie die Aufzeichnung beenden. Wenn Sie wieder an dieselbe Stelle im Programm springen müssen, spielen Sie einfach das Makro ab.

Create

[Alt][=]

Mit dem Befehl *Create* werden Tastaturanschlüge für eine bestimmte Makrotaste aufgenommen. Das Tastenkürzel für *Create* lautet [Alt][=].

Anschließend werden Sie gefragt, welcher Tastenkombination das Makro zugewiesen werden soll. Drücken Sie die entsprechende Taste(nkombination), zum Beispiel [Alt][M]. Daraufhin erscheint die Statusmeldung *Recor-*

ding in der rechten oberen Ecke des Bildschirms, solange Sie das Makro aufzeichnen.

Stop recording
[Alt][-]
Mit dem Befehl *Stop recording* wird die Aufnahme der Tastendrucke eines Makros beendet. Verwenden Sie diesen Befehl, nachdem Sie *Options/Macros/Create* ausgewählt haben. Das Tastenkürzel für *Stop recording* lautet [Alt][-].

Achtung
Verwenden Sie nicht *Options/Macro/Stop Recording*, um die Makro-Aufzeichnung zu beenden; diese Befehlsfolge würde in das Makro aufgenommen werden!

Remove
Der Befehl *Remove* entfernt das Makro einer einzelnen Taste(nkombination). Nach dem Aufruf dieses Befehls werden Sie nach der entsprechenden Taste(nkombination) gefragt. Wählen Sie das Makro an, das Sie löschen wollen, und drücken Sie die Taste [Eingabe].

Delete All
Mit dem Befehl *Delete ALL* werden alle Makrodefinitionen gelöscht. Alle Tasten und Tastenkombinationen erhalten wieder ihre ursprüngliche Bedeutung.

Makros aufzeichnen

Um ein Makro aufzuzeichnen,

1. wählen Sie den Befehl *Options/Macros/Create* (oder [Alt][=]), um mit der Makrodefinition zu beginnen. Sie werden nach der Taste gefragt, der Sie das Makro zuordnen wollen,
2. wählen Sie eine Taste aus, die noch keine Zuordnung hat, wie [Umschalt][F10],
3. wenn Sie schließlich mit der Makroaufzeichnung beginnen, werden alle im Folgenden eingegebenen Tasten in das Makro aufgenommen (allerdings nicht die Tasten, die als Programmeingabe gedrückt werden und Mausbewegungen),
4. beenden Sie die Makroaufzeichnung; drücken Sie [Alt][-] oder die Tastenkombination, die Sie für dieses Makro reserviert haben (in diesem Beispiel [Umschalt][F10]),
5. speichern Sie das Makro in einer Konfigurationsdatei: Wählen Sie dazu den Befehl *Options/Save Options* und aktivieren Sie das Markierungsfeld *Macros* im Dialogfenster *Save Configuration*. Geben Sie den Namen der jeweiligen Konfigurationsdatei ein (wenn der Name im Dialogfenster noch nicht aufgeführt ist), anschließend wählen Sie OK.
6. setzen Sie die Analyse des Programms fort.

Display Options

Abbildung 2.19:
Das Dialogfenster
Display Options

Der Befehl *Options/Display Options* öffnet das nachfolgend dargestellte Dialogfenster *Display Options*:



Mit dem Dialogfenster *Display Options* können Sie folgende Einstellungen vornehmen:

- Sollen der Inhalt des Bildschirmspeichers während des Programmlaufs ausgelagert werden?
- Mit welchem Spaltenabstand werden Tabulatoren im *Module*-Fenster angezeigt?
- 25-Zeilen oder 43/50-Zeilen für die Bildschirmdarstellung von Turbo Profiler?
- Wie breit können die Routinennamen in den Fenstern *Execution Profile*, *Callers* und *Areas* sein, damit sie noch vollständig dargestellt werden können?

Display Swapping

Das Schaltfeld *Display Swapping* bietet zwei Optionen für die Umschaltung zwischen dem Ausgabebildschirm des Programms und der Benutzerumgebung von Turbo Profiler: *None* und *Always*.

- *None* bedeutet, daß zwischen den beiden Bildschirmen niemals umgeschaltet wird.

Verwenden Sie diese Option, wenn das untersuchte Programm keine Meldungen auf dem Bildschirm ausgibt.

- *Always* bedeutet, daß jedesmal, wenn das Programm läuft, auf den Ausgabebildschirm umgeschaltet wird.

Verwenden Sie diese Option, wenn Sie die Bildschirmausgaben Ihres Programms sehen möchten.

Wenn Sie eine Windows-Anwendung analysieren, wird *Display Swapping* automatisch auf *Always* gesetzt und kann nicht geändert werden.

Screen Lines

In *Screen Lines* geben Sie an, ob Turbo Profiler die normale 25-Zeilendarstellung oder die 43-/50-Zeilendarstellung verwenden soll, wie sie auf EGA und VGA-Systemen möglich ist.

Je nachdem, welche Art von Bildschirmadapter Sie besitzen, ist nur eine oder sind beide Optionen verfügbar. Bei Monochrom- oder CGA-Adaptoren ist nur der 25-Zeilenmodus möglich.

Tab Size

Im Eingabefeld *Tab Size* legen Sie den Spaltenabstand für Tabulatoren zwischen 1 und 32 fest. Sie können die Breite der Tabulatorspalten verringern, um mehr vom Text in den Quelldateien zu sehen, die mit vielen Tabulatoren eingerückt sind.

Width of Names

Im Eingabefeld für die Länge der Namen geben Sie an, bis zu welcher Länge die Routinenbezeichnungen in den Fenstern *Execution Profile*, *Callers* und *Areas* vollständig dargestellt werden können.

Path for Source

Standardmäßig sucht Turbo Profiler die Quelldateien Ihres Programms an folgenden Stellen in der angegebenen Reihenfolge:

1. im Verzeichnis, in dem der Compiler die Quelldateien vorfand;
2. in den Verzeichnissen, die Sie (evtl.) unter *Options/Path for Source* oder unter Verwendung der Kommandozeilenoption **-sd** angegeben haben.
3. im aktuellen Verzeichnis;
4. im Verzeichnis, in dem sich die Programmdatei befindet.

Im Befehl *Options/Path for Source* können Sie eine Liste von Verzeichnissen angeben, die der Compiler vor dem aktuellen Verzeichnis zusätzlich durchsuchen soll.

Geben Sie die zu durchsuchenden Verzeichnisse in folgender Art und Weise ein:

Verzeichnis; Verzeichnis; Verzeichnis

Zum Beispiel:

C:\PROGRAMS;C:\BC\BIN

Save Options

Mit dem Befehl *Options/Save Options* werden alle eingestellten Optionen in einer Konfigurationsdatei gespeichert. Wenn Sie die gespeicherten Profiler-Optionen später wieder laden wollen, können Sie diese Konfigurationsdatei mit dem Befehl *Options/Restore Options* wieder einlesen.

Nach Auswahl von *Options/Save Options* erscheint das Dialogfenster *Save Configuration*:

Abbildung 2.20:
Das Dialogfenster
Save Configuration



In diesem Dialogfenster können Sie alle momentan eingestellten Optionen, die Anordnung der Fenster und die Makros in einer Datei speichern. *Options*, *Layout* und *Macros* sind als Markierungsfelder angegeben. Das bedeutet, daß Sie eine, zwei oder alle drei Arten von Informationen in einer Konfigurationsdatei speichern können.

- *Options* sind Menüpunkte, die nicht in einer .TFA- oder .TFS-Datei gespeichert werden (wie *Options/Path for Source*, Kommandozeilenoptionen und die Einstellungen im Dialogfenster *Display Options*).
- In *Layout* werden alle momentan geöffneten Fenster mit Reihenfolge, Position und Größe berücksichtigt.
- Mit *Macros* werden die Tastaturmakros gespeichert.
- Unter *Save To* ist die standardmäßige Konfigurationsdatei, TFCONFIG.TF, vorgegeben. Um Ihre Optionen in dieser Datei zu speichern, wählen Sie *OK* aus (oder drücken [Eingabe]).

Um die Optionen in einer anderen Datei zu speichern, geben Sie einen anderen Dateinamen (optional mit Laufwerk und Pfad) an und wählen *OK* (oder drücken Sie [Eingabe]).

Nachdem die Optionen in einer Konfigurationsdatei gespeichert wurden, können Sie die Einstellungen jederzeit mit dem Befehl *Options/Restore* wieder einlesen.

Restore Options

Mit dem Befehl *Options/Restore* können Sie Ihre Einstellungen für die Analyse aus einer Datei einlesen. Es lassen sich auch mehrere verschiedene Konfigurationsdateien verwenden, in denen unterschiedliche Makros, Fensteranordnungen, usw. gespeichert sein können.

Nach Auswahl von *Options/Restore* erscheint das Dialogfenster *Restore*. Das Dialogfenster *Restore* funktioniert wie alle anderen Dialogfenster des Profilers, mit denen Dateien geladen werden können:

- Geben Sie einen Dateinamen oder eine Dateiauswahl (mit DOS-Jokerzeichen) im Eingabefeld *File name* an.
- Wählen Sie ein anderes Laufwerk oder einen anderen Pfad im Verzeichnisbaum aus.
- Wählen Sie einen Dateinamen in der Dateiliste aus.

- Schließen Sie den Vorgang mit *OK* ab (oder verlassen Sie das Dialogfenster, ohne eine Datei zu laden, indem Sie *Cancel* auswählen).
- Mit *Help* können Sie ein Fenster mit Hilfestellungen zur Benutzung dieses Dialogfensters öffnen.

Nachdem Sie den Namen einer Konfigurationsdatei eingegeben oder ausgewählt haben und die Datei geladen wurde, übernimmt Turbo Profiler alle gespeicherten Optionen, Einstellungen, die Fensteranordnung und Makros in die aktuelle Benutzerumgebung. (Es können nur Konfigurationsdateien eingelesen werden, die mit dem Befehl *Options/Save Options* angelegt wurden.)

Window-Menü

Mit den Befehlen im Menü *Window* können Sie

- die Fenstergrößen von Turbo Profiler verändern;
- innerhalb und zwischen den Fenstern hin- und herspringen;
- Fenster in Symbole (*icons*) umwandeln und wieder in Fenster zurückverwandeln;
- Fenster schließen und wieder öffnen;
- auf den Ausgabebildschirm Ihres Programms umschalten;
- ein offenes Fenster aktivieren.

Die Befehle im oberen Abschnitt des Menüs *Window* dienen der Bewegung innerhalb der Fensterumgebung und der Anordnung einzelner Fenster nach Ihren Bedürfnissen. Die meisten Fenster von Turbo Profiler enthalten bestimmte Standardelemente, die in allen Fenstern gleich sind (Rollbalken, Schließfeld, Zoom-Feld, usw.). Diese Elemente sind zu Beginn dieses Kapitels ausführlich beschrieben.

Zoom

Der Befehl *Zoom* vergrößert das aktive Fenster (erkennbar an der doppelten Umrahmung) auf die gesamte Bildschirmfläche.

Next

Mit dem Befehl *Next* wird das Fenster mit der nächsthöheren Nummer aktiviert.

Next Pane

In Fenster mit mehreren Ausschnitten wechselt Sie mit dem Befehl *Next Pane* in den nächsten Ausschnitt.

Size/Move

Der Befehl *Size/Move* aktiviert den Modus zur räumlichen Organisation der Fenster. Sie können das aktuelle Fenster mit den Pfeiltasten [*Nach oben*], [*Nach unten*], [*Nach rechts*] und [*Nach links*] verschieben. In Kombination mit der [*Umschalt*]-Taste läßt sich das Fenster mit den Pfeiltasten vergrößern und verkleinern. In der Statuszeile finden Sie eine Erläuterung, welche Tastenkombinationen welche Vorgänge bewirken. Das Tastenkürzel für diesen Befehls lautet [*Strg*][*F5*].

Iconize/Restore

Der Befehl *Iconize/Restore* erlaubt die Verkleinerung des aktiven Fensters in ein Symbol (*icon*) sowie den umgekehrten Vorgang der Umwandlung eines Symbols in ein Fenster.

Die Symbolfunktion von Turbo Profiler stellt eine sehr praktische Methode dar, um mehrere Fenster geöffnet zu halten, ohne daß der Bildschirm unübersichtlich wird. Ein Fenstersymbol ist eine kleine Darstellung eines geöffneten Fenster

Um ein Fenster in ein Symbol umzuwandeln, wählen Sie den Befehl *Iconize/Restore* im *Window*-Menü aus oder klicken das Symbolfeld im oberen Rahmen des Fensters an. Für die Rückverwandlung eines Symbols in die ursprüngliche Darstellung wählen Sie erneut den Befehl *Iconize/Restore* oder klicken auf die Zoom-Box des Symbols.

Close

Das Kommando *Close* schließt zeitweise das aktuelle Fenster. Um das Fenster wieder herzustellen, wählen Sie *Undo Close*.

Undo Close

Der Befehl *Undo Close* öffnet das zuletzt geschlossene Fenster. Gleichzeitig wird dieses Fenster aktiviert.

User Screen

Wählen Sie den Menüpunkt *Window/User Screen* (oder drücken Sie *[Alt][F5]*, um den Ausgabebildschirm Ihres Programms in voller Größe zu sehen. Mit jeder beliebigen Taste kommen Sie wieder in die Profiler-Umgebung zurück.

Liste der geöffneten Fenster

Am unteren Ende des *Window*-Menüs finden Sie eine numerierte Liste der momentan geöffneten Fenster. Drücken Sie die entsprechende Nummer des Fensters, das aktiviert werden soll.

Das Menü Help

Über das Menü *Help* können Sie online auf die Hilfestellungen in einem speziellen Fenster zugreifen. Die Hilfsinformationen behandeln fast jeden Aspekt der Benutzerumgebung und Bedienung von Turbo Profiler. (Außerdem erscheinen zu jeder Auswahl eines Befehls einzeilige Menü- und Dialog-Hinweise in der Statuszeile.)

Zum Öffnen des Help-Menüs haben Sie zwei Möglichkeiten:

- Sie können jederzeit *[F1]* oder *[Alt][F1]* drücken (auch wenn Sie sich in einem Dialogfenster befinden oder ein Menübefehl ausgewählt ist).
- Klicken Sie auf *Help*, wenn Sie in der Statuszeile oder in einem Dialogfenster angegeben ist.

Mit *[Esc]*, durch Anklicken des Schließfeldes oder den Befehl *Window/Close* können Sie das *Help*-Fenster wieder schließen.

Viele Hilfsinformationen enthalten *Stichworte*, die hervorgehoben dargestellt sind. Wählen Sie diese Stichworte mit den Pfeiltasten und *[Eingabe]*, wenn Sie weitere Informationen wünschen. Mit den Tasten *[Pos1]* und *[Ende]* können Sie zum ersten bzw. letzten Stichwort auf dem Bildschirm springen. Mit der Maus ist es noch einfacher: klicken Sie einfach das gewünschte Stichwort an, um den Hilfetext zu erhalten.

Index

Der Befehl *Help/Index* öffnet ein Dialogfenster, in dem eine vollständige Liste aller Stichwörter (die in den Hilfsinformationen hervorgehoben dargestellt werden) enthalten ist.

Sie können die Liste mit der Taste *[Bild unten]* durchblättern. Wenn Sie das gesuchte Stichwort gefunden haben, wählen Sie es mit den Pfeiltasten aus und drücken *[Eingabe]* (oder führen einen Doppelklick aus).

Previous Topic

Der Befehl *Help/Previous Topic* öffnet das Fenster mit den Hilfsinformationen und stellt den Text dar, der zuletzt betrachtet wurde.

In Turbo Profiler können Sie auf die letzten zwanzig Hilfsbildschirme zurückgreifen. Mit der in der Statuszeile angezeigten Taste *[Bild unten]* können Sie zum jeweils letzten Hilfebildschirm zurückkehren.

Help on Help

Der Befehl *Help/Help on Help* öffnet einen Textbildschirm, in dem die Bedienung des Hilfesystems in Turbo Profiler erklärt ist.

Analyse-Strategien

Die Verbesserung des Laufzeitverhaltens eines Programms ist kein geradliniger Vorgang. Es wird wahrscheinlich nicht so sein, daß Sie ein Programm untersuchen, den Quellcode ändern und damit fertig sind. Die Untersuchung eines Programms im Hinblick auf ein besseres Laufzeitverhalten mit Hilfe des Turbo Profiler ist ein dynamischer und interaktiver Vorgang. Sie erfassen statistische Daten, analysieren die Ergebnisse in einer Reihe verschiedener Fenster und ändern vielleicht auch einige der Untersuchungsparameter, um andere statistische Werte zu erhalten, anschließend werten Sie diese Ergebnisse wieder aus, ändern den Quellcode, führen eine erneute Messung durch, usw.

Wenn Sie anfangs noch nicht wissen, wo sich die Engpässe in Ihrem Programm befinden, sollten Sie die Standard-Einstellungen von Turbo Profiler verwenden. Beim Betrachten der Ergebnisse im Fenster *Execution Profile* erkennen Sie, welche Routinen in Ihrem Programm die meiste Zeit beanspruchen. Durch die Gegenüberstellung der Daten über Zeitdauer und Anzahl der Aufrufe, können Sie herausfinden, welche Programmteile in Bezug auf Zeitbedarf *pro Aufruf* am aufwendigsten sind. Mit diesem Wissen können Sie damit beginnen, die Problembereiche in Ihrem Programm zu entfernen.

Turbo Profiler stellt einige unterschiedliche Report-Fenster zum Auswerten der erfaßten Daten zur Verfügung. Den Inhalt dieser Fenster können Sie wahlweise auf Papier ausdrucken oder in eine Datei schreiben, um die Leistungssteigerung zu dokumentieren. In den Report-Fenstern finden Sie Angaben über Zeitdauer und Anzahl der Ausführung Ihrer Programmteile, Datei-Zugriffe, DOS-Interrupts und Overlay-Zugriffe sowie ein Protokoll der Aufrufe Ihrer Routinen.

Was können Sie mit all diesen Möglichkeiten machen? Wie können Sie Turbo Profiler für eine effektive und effiziente Analyse einsetzen? Und

welche Kniffe gibt es für die Analyse? Es ist klar, daß wir nicht alle diese Fragen in diesem Kapitel beantworten können. Wir wollen Ihnen aber einige allgemeine Richtlinien, Methoden und Strategien für den Anfang mitgeben.

Wenn Sie mit Turbo Profiler zum ersten Mal ein Programm starten:

- wird der Profile-Modus auf „Active“ gesetzt;
- wird Ihre .EXE-Datei nach dem Hauptmodul durchsucht;
- werden Bereichsmarkierungen für das Programm gesetzt;
- wird festgestellt, welches Quellmodul das Hauptprogramm enthält;
- wird das Hauptmodul in das Module-Fenster geladen;
- wird der Cursor an den Anfang des Hauptmoduls gesetzt.

Das Hauptmodul ist in der Regel das Modul, das die erste ausführbare Quellcode-Zeile Ihres Programms enthält. Ausnahmen bilden Initialisierungsteile in Pascal-Units, die vor der BEGIN-Anweisung des Hauptmoduls in der Reihenfolge ihres Imports ausgeführt werden. Bereichsmarkierungen sind die Stellen, an denen Statistiken erfaßt werden. Die Anzahl der Markierungen hängt von der Anzahl an Symbolen ab, die in den Debug-Informationen Ihres Programms enthalten sind.

Hinweis

Jedesmal, wenn Sie Turbo Profiler verlassen, werden die *Informationen* über die eingestellten Bereiche für das gerade geladene Programm in einer *Bereichsdatei* gespeichert. Der Name dieser Datei lautet *Dateiname.TFA*, wobei als *Dateiname* der Name des Programms verwendet wird. Wird ein Programm geladen, sucht Turbo Profiler jedesmal nach der entsprechenden .TFA-Datei im Zugriffspfad. Wenn eine vorhanden ist, wird sie automatisch für die Einstellung der Bereiche in dieser Datei verwendet.

Es empfiehlt sich, die Ergebnisse einer Analyse zu speichern, vor allem dann, wenn die Analyse sehr langwierig war oder eine weitere Auswertung zu einem späteren Zeitpunkt vorgenommen werden soll.

Mit dem Befehl *Statistics / Save* können Sie die Ergebnisse einer Analyse auch in einer .TFS-Datei speichern. Standardmäßig erhält die Statistik-Datei den Namen *Dateiname.TFS*. Sie können diese Standardeinstellung verwenden oder den Namen ändern (falls Sie mehrere Sätze von Meßdaten für ein Programm speichern wollen).

Vorbereitungen für die Analyse

Die Beispiele in Kapitel 1 sind klein und einfach. Sie sind dazu gedacht, den allgemeinen Ablauf einer Analyse zu demonstrieren. Die Aufgabe dieses Kapitels war die Optimierung der Routine **prime** und nicht das Aufzeigen spezieller Engpässe in einem Programm.

Sie brauchen jedoch einen Profiler viel eher beim Erstellen umfangreicherer Programme. Denn zuerst müssen Sie herausfinden, welche Programmtteile einen Engpaß darstellen, *danach* können Sie sich darüber Gedanken machen können, welche Konsequenzen Sie aus Ihren Erkenntnissen ziehen wollen.

Bevor Sie Ihr Programm untersuchen, sollten Sie Ihren Quellcode so organisieren, daß die erfaßten Statistiken brauchbar sind. Sobald der Quellcode vorbereitet ist, compilieren Sie das Programm so, daß Debug-Informationen enthalten sind. Anschließend setzen Sie die Markierungen, damit der Profiler erkennen kann, wo eine Statistik und welche Art von Statistik erfaßt werden soll.

Die Anpassung Ihres Programms

Als erstes müssen Sie Ihr Programm also so vorbereiten, daß die gewünschten Informationen ermittelt werden können. Wenn Sie zum Beispiel ein interaktives Programm schreiben, das viele Eingaben von der Tastatur erhält, brauchen Sie sich nicht zu wundern, daß Ihr Programm die meiste Zeit damit verbringt, auf die Eingaben des Anwenders zu warten.

Überlegen Sie sich vorher, auf welche Weise Sie sich die statistischen Daten besorgen wollen und richten Sie Ihren Quellcode so ein, daß Sie mit diesen Daten auch etwas anfangen können. Wenn Sie dann den Quellcode entsprechend modifiziert haben (falls es nötig war), compilieren Sie das Programm erneut – unter Verwendung der Debug-Informationen. Anschließend setzen Sie Bereichsmarkierungen, um dem Compiler mitzuteilen, wo er statistische Daten sammeln soll und welche Arten von Statistiken zu erstellen sind.

Einige grundsätzliche Methoden, um Engpässe in großen Programm aufzuspüren:

- Wählen Sie Bereiche, die groß genug sind, um eine vernünftige Analyse durchzuführen.

Wenn Sie ein Programm zum Suchen von Strings auf eine dreizeilige Datei anwenden, wird Ihnen das wenig Aufschluß liefern. Einen kurzen String zu suchen, der in fast jeder Zeile einer Datei mit 10000 Zeilen vorkommt, oder einen langen String zu suchen, der in 10000 Zeilen nur an einer Stelle steht, führt ebenfalls zu Ergebnissen, die nicht mehr miteinander vergleichbar sind.

- Wenn Sie wissen, daß Ihr Programm sehr schnell ist, stellen Sie den Profiler so ein, daß mehrere Programmläufe automatisch hintereinander ausgeführt werden. (Ändern Sie dazu die Einstellung von *Run Count* im Dialogfenster von *Statistics / Profiling options*.)

- Ändern Sie Ihr Programm so, daß keine Tastatureingaben erforderlich sind, oder entfernen (bzw. deaktivieren) Sie alle Bereiche, in denen Tastatureingaben angefordert werden.

Lesen Sie die Daten von einer Datei ein oder verwenden Sie einen Zufallsgenerator, um Zahlen in ein Array zu schreiben. Es kommt darauf an, daß Sie Daten auswählen, die für die praktische Anwendung dieses Moduls typisch sind.

- Isolieren Sie die Module des Programms, von denen Sie wissen, daß sie verbessert werden müssen.

Compilierung des Programms

Nachdem Sie Ihr Programm so vorbereitet haben, daß die Analyse nicht zu einem Rätselraten wird, compilieren Sie das Programm erneut. Dateien, die so compiliert wurden, daß sie mit Turbo Debugger untersucht werden können, brauchen nicht neu compiliert zu werden.

Turbo Profiler verarbeitet 16-bit-Programme von Turbo C++, sowie Programme von Borland C++ und Turbo Assembler. Achten Sie beim Compilieren darauf, daß der .EXE-Datei alle symbolischen Debug-Informationen hinzugefügt werden.

- **Borland C++** Markieren Sie im Dialogfenster *Projektoptionen* die folgenden zwei Optionen:
 - *Compiler / Debugger / Debug-Information in OBJ-Dateien.*
 - *Linker / Allgemein / Debug-Informationen mit aufnehmen.*
- **Turbo Assembler:** Der Quellcode muß mit der Kommandozeilenoption **/zi** assembliert und mit TLINK über die Option **/v** gelinkt werden.

Turbo Profiler benötigt zur Analyse sowohl die .EXE-Datei als auch die ursprünglichen Quelldateien. Die Quelldateien werden in den folgenden Verzeichnissen in der angegebenen Reihenfolge gesucht:

1. In dem Verzeichnis, in dem sich die Datei zum Zeitpunkt der Compilierung befand (diese Information steht in der Programmdatei);
2. In den Verzeichnissen, die mit *Options / Path* angegeben wurden (oder in den Verzeichnissen, die mit Hilfe der Kommandozeilenoption **-sd** vorgegeben wurden)
3. Im aktuellen Verzeichnis;
4. In dem Verzeichnis, in dem sich die zu untersuchende .EXE-Datei befindet.

Bereiche festlegen

Nachdem alle Programmvorbereitungen soweit abgeschlossen sind, daß Sie sich auf die Problembereiche konzentrieren können, sind Sie bereit, den Profiler zu starten und Statistiken für die einzelnen Bereiche zu ermit-

teln. Sie können damit beginnen, das gesamte Programm allgemein zu untersuchen und sich dann nach und nach zu den Problembereichen vor-tasten. Übernehmen Sie am Anfang die Standardbereiche – Turbo Profiler setzt die Standardbereiche entsprechend der Dichteverteilung der Symbole in der Programmdatei.

Ein *Bereich* ist, wie Sie sich erinnern, ein Programmteil, an dem Sie eine Messung vornehmen wollen. Ein Bereich kann eine einzelne Zeile, eine Schleife oder eine ganze Routine sein. Eine *Bereichsmarkierung* setzt einen internen Breakpoint. Jedesmal, wenn der Profiler an einen dieser Breakpoints gelangt, wird ein bestimmter Code ausgeführt – je nachdem, welche Optionen für den fraglichen Bereich gesetzt sind. Der Profiler-Code kann eine komplizierte Routine zum Erfassen einer Vielzahl von Informationen oder ein einfacher Befehl sein, um die Programmausführung anzuhalten.

Folgende Operationen können durchgeführt werden, wenn der Profiler in einen Bereich gelangt:

Operation	Beschreibung
<i>all</i>	Aktiviert die standardmäßige Zählung (Erfassung von Zeitdauer und Anzahl der Ausführung eines markierten Bereichs).
<i>Enable</i>	Schaltet die Erfassung der Statistiken ein (falls sie zuvor abgeschaltet wurde).
<i>Disable</i>	Schaltet die Erfassung der Statistiken ab, ohne das Programm abubrechen. Wenn Ihr Programm einen Bereich erreicht, in dem die Erfassung der Statistiken mit <i>Enable</i> eingeschaltet ist, setzt der Profiler die Erfassung wieder fort.
<i>Stops</i>	Hält das Programm an und übergibt die Programmsteuerung an Turbo Profiler (oder an DOS, wenn Sie im Stapelverarbeitungs-Modus arbeiten). An dieser Stelle können Sie die erfaßten Daten untersuchen und anschließend die Ausführung des Programms wieder fortsetzen.

Standardmäßig zählt Turbo Profiler die Anzahl der Ausführungen eines Bereichs und die Zeitdauer, die das Programm in diesem Abschnitt verweilt. Sie können die Vorgehensweise von Turbo Profiler für einen Bereich ändern, indem Sie die Option *Operation* im Dialogfenster *Area Options* verändern; dieses Dialogfenster erhalten Sie über die lokalen Menüs der Fenster *Module* oder *Areas*.

Wenn Sie vor dem Start einer Analyse die Bereiche in Ihrem Programm festlegen, sollten Sie folgende Fragen beantworten:

- In wievielen Bereichen sollen Daten erfaßt werden?
- Welche Abschnitte des Programms sollen untersucht werden?
- Was soll in den einzelnen markierten Bereichen passieren?

Wie detailliert soll die Analyse sein?

Es gilt zunächst zu entscheiden, wieviele Informationen Sie benötigen. Bedenken Sie, wie groß Ihr Programm ist und wie lange die Ausführung des Programms wohl dauern wird:

- Bei einem kleinen Programm werden Sie wahrscheinlich jede Programmzeile untersuchen – die detaillierteste Variante.
- Bei großen Programmen sind weniger Einzelheiten erforderlich. Die Ermittlung des Zeitbedarfs der einzelnen Routinen ist wahrscheinlich ausreichend.

Die Angabe „groß“ ist ein wenig ungenau: Entscheidend ist die Anzahl der Module, Routinen und Quellcode-Zeilen.

Wenn sich Ihr Quellcode aus 10000 Zeilen in 10 Modulen zusammensetzt, sollten Sie nur jeweils ein Modul in einer aktiven Analyse untersuchen. Dazu sollte Ihr Programm in funktionell eigenständige Module aufgeteilt sein.

Wenn Ihr Programm nicht mehr als 100 Zeilen umfaßt und eine detaillierte Analyse erforderlich ist, werden Sie wahrscheinlich über alle Zeilen eine Messung vornehmen.

Falls Ihr Programm in weniger als fünf Sekunden abgearbeitet wird, erhalten Sie genauere Ergebnisse für Ihre Analyse, wenn Sie mehrere Programmläufe durchführen und den Durchschnitt der Ergebnisse heranziehen. (Geben Sie die Anzahl an Programmläufen mit dem Befehl *Statistics / Profiling* an.) Wenn die Ausführung des Programms (ohne den zusätzlichen Aufwand für die Analyse) eine Stunde erfordert, sollten Sie mit der Festlegung von Bereichen sparsam sein, damit die Ausführungszeit nicht über alle Maßen ansteigt.

Weitere Bereiche in die Analyse aufnehmen

Teilen Sie Ihr Programm mit dem Befehle *Add Areas* im lokalen Menü des *Module*-Fensters in eine Reihe von Bereichen auf und starten Sie dann das Programm, um die einzelnen Bereiche zu messen.

Wenn Sie Ihr Programm nicht unterteilen, wird es nach einem Standardschema in geeignete Bereiche unterteilt. Ausgehend von der Symboltabelle des Programms wählt Turbo Profiler eine von mehreren Standardoptionen für die Einstellung der Bereiche in Ihrem Programm aus:

- Wenn die Tabelle nur wenige Symbole enthält und nur ein Modul vorliegt, verwendet Turbo Profiler *Every Line* als Standardeinstellung.
- Wenn viele Symbole und mehrere Module vorkommen, wählt Turbo Profiler *All Routines* als Standardeinstellung für die Bereiche.

Hinweis

Wenn Ihr Programm sehr groß ist, sollten Sie es zuerst im passiven Modus analysieren, um einen groben Überblick zu gewinnen. Sie können dann später Bereiche für eine detailliertere Analyse auswählen.

Welche Daten wollen Sie untersuchen?

Turbo Profiler sammelt die folgenden Standardinformationen über jeden Bereich Ihres Programms (Aktiv-, als auch Passivmodus):

- die Anzahl der Ausführungen dieses Bereichs;
- die Zeit, die das Programm in diesem Bereich zubringt (aktiver Modus);
- die Anzahl der Zeittakte während der Ausführung dieses Bereichs (passiver Modus).

Es lassen sich aber auch noch einige weitere Informationen in einer Analyse erfassen:

- Durch die Optionen *Statistics / Callers* und *Call Stack* im Dialogfenster *Area Options* können Sie verfolgen, von welchen Routinen eine markierte Routine aufgerufen wird – wie oft und über welchen Aufrufpfad.
- Mit der Option *Statistics / Files* können Sie die Dateizugriffe in Ihrem Programm überwachen.
- Die Option *Statistics / Interrupts* ermöglicht die Aufzeichnung aller Interrupts in Ihrem Programm.
- Zugriffe auf Overlays lassen sich durch Einschalten der Option *Statistics / Overlays* festhalten.

Nachdem Sie die gewünschten Optionen im Menü *Statistics* ausgewählt haben, können Sie das entsprechende Report-Fenster öffnen (mit den Befehlen im Menü *Views*) und die lokalen Menüs der einzelnen Fenster aufrufen, um festzulegen, welche Einzelheiten bei der Erfassung der Daten berücksichtigt werden sollen.

Um die gewünschten Reports von Turbo Profiler zu erhalten, müssen Sie die entsprechenden Optionen setzen, *bevor* Sie das Programm ausführen.

Wann soll die Datenerfassung beginnen?

Häufig wollen Sie die Zeitdauer nur für einen bestimmten Programmabschnitt erfassen. Dazu starten Sie das Programm ohne Datenerfassung, indem Sie *Statistics / Accumulation* auf *Disabled* setzen. Sie können die Einstellung der Option *Accumulation* jederzeit ermitteln, indem Sie das Dialogfenster *File / Get Info* aktivieren und den Status von *Collection* überprüfen.

Wenn *Accumulation* ausgeschaltet (*Disabled*) ist, müssen Sie die Bereichsmarkierung für den Bereich, ab dem mit der Erfassung begonnen werden soll, auf *Enable* setzen, und die Markierung für den Bereich, ab dem die Erfassung wieder abgeschaltet werden soll, auf *Disable* einstellen. Die Anzahl der Anfangs- und Endpunkte ist nur durch den verfügbaren Speicher beschränkt. Im allgemeinen können Sie so viele setzen, wie Sie benötigen.

Wie sollen die Zeitdaten zusammengefaßt werden?

Der Profiler kann die Ausführungszeit von Unterprogrammen in der aufrufenden Routine zusammenfassen oder auch getrennt erfassen.

Standardmäßig legt der Profiler die aufrufende Routine auf den Stack und schaltet die Erfassung für die aufrufende Routine ab, falls die aufgerufene Routine eine Bereichsmarkierung aufweist. Der Profiler ordnet alle Zeittakte der aufgerufenen Routine und nicht der aufrufenden zu, solange sich das Programm in der aufgerufenen Routine befindet.

Falls Sie jedoch angeben, daß die aufrufende Routine eine kombinierte Zeiterfassung (anstelle einer getrennten Zeiterfassung) vornehmen soll, wird die Zeitdauer in der aufgerufenen Routine der aufrufenden Routine hinzugefügt.

Im Standard-Modus von Turbo Profiler wird eine eigene Zeiterfassung für jede markierte Routine vorgenommen. Der Zeitverbrauch einer Routine wird deshalb normalerweise ohne Berücksichtigung der Zeiten gemessen, die für den Aufruf ihrer Unterprogrammen Routinen anfallen. Wenn Sie in den Zeitdaten einer Routine die Ausführungszeit ihrer Unterprogrammen Routinen berücksichtigen wollen, wählen Sie die Einstellung *Combined* für die Option *Timing* im entsprechenden Dialogfenster des Fensters *Areas*.

Welche Daten wollen Sie sehen?

Es ist sehr wichtig zu wissen, wie man die Informationsmenge, die Turbo Profiler erfaßt und anschließend anzeigt, steuern kann – vor allem, wenn Sie spezielle Informationen aus einem Abschnitt eines großen Programms ermitteln wollen. Turbo Profiler bietet zwei Möglichkeiten, um festzulegen, wieviele Informationen über Ihr Programm ausgegeben werden:

- Bevor Sie mit der Analyse beginnen, können Sie die Erfassung auf bestimmte Bereiche und Datenarten durch die Einstellung von Optionen und Parametern beschränken.
- Nach der Analyse lassen sich die erfaßten Daten (ohne daß sie gelöscht werden) filtern, so daß nur diejenigen Daten angezeigt werden, für die Sie sich im Moment interessieren.

In den Fenstern *Module*, *Areas* und *Interrupt* können Sie angeben, für welche Programmteile Daten erfaßt und wieviele Informationen dort erfaßt werden sollen. Sie können die Datenerfassung so grob einstellen, daß nur die Routinen eines Moduls erfaßt werden, oder Sie stellen die Datenerfassung so fein ein, daß jede einzelne Anweisung separat berücksichtigt wird. Es ist auch möglich, sich auf die reinen Zeitdaten zu beschränken (indem man eine passive Analyse durchführt) oder alle verfügbaren Daten mit allen Aufruf-Hierarchien, Datei- und Overlay-Zugriffen sowie DOS- und BIOS-Interrupts zu erfassen. Der Zeittakt des Profilers läßt sich beschleunigen oder verlangsamen, wodurch sich die Auflösung der erfaßten Daten beliebig einstellen läßt (nur im passiven Modus).

Hinweis

Die Menge der erfaßten Daten wirkt sich wie folgt aus: Je mehr Informationen Turbo Profiler erfaßt, desto langsamer läuft das Programm und desto mehr Speicherplatz wird benötigt, um die erfaßten Daten zu speichern.

Weitere Informationen über das Filtern der angezeigten Daten finden weiter unten im Abschnitt „Filtern der erfaßten Daten“.

Nachdem alle Daten erfaßt sind, können Sie Befehle in den Report-Fenstern verwenden, um Daten herauszufiltern, die Sie zwischenzeitlich nicht betrachten wollen.

Die Analyse Ihres Programms

Nachdem die Bereiche bestimmt wurden, können Sie mit der Analyse beginnen. Sie können das Ergebnis mit dem Befehl *Statistics / Save...* speichern. Dieser Befehl speichert die Statistiken in eine .TFS-Datei (Turbo Profiler-Statistik). Wenn Sie mehrere verschiedene Analysen speichern wollen, sollten Sie die Namen der Dateien nach einer Konvention vergeben, durch die jeder Lauf eindeutig zugeordnet werden kann (zum Beispiel RUN1.TFS, RUN2.TFS, usw.). Dadurch können Sie die Statistiken später leichter vergleichen.

Unter Umständen wissen Sie nämlich nicht sofort, ob es sich lohnt, eine Analyse aufzubewahren, bis Sie verschiedene statistische Ergebnisse miteinander verglichen haben.

Nach dem Speichern der .TFS-Datei können Sie die Ergebnisse der Analyse in den Report-Fenstern studieren, sortieren und aus den Daten die interessanten Ergebnisse für die Anzeige herausfiltern, um die Ergebnisse zu deuten. Sie verlieren keine der gesetzten Bereichsmarkierungen oder statistischen Reports, da all diese Informationen wiederhergestellt werden können (sie stehen alle in der .TFS-Datei). Eine langwierige Analyse sollten Sie im allgemeinen immer abspeichern, falls Sie nicht absolut sicher sind, daß Sie die Ergebnisse nicht mehr brauchen.

Was wollen Sie herausfinden?

Normalerweise wird ein Profiler eingesetzt, um eine Antwort auf eine der nachfolgenden Fragen zu erhalten:

- Wie effizient ist der Algorithmus? (*Testen von Algorithmen*)
- Macht das Programm alles, was es soll? Läuft überhaupt alles? (*Verifizierung und Testen*)
- Wie hoch sind die Laufzeiten der einzelnen Routinen? Wieviel Zeit wird für die einzelnen Ressourcen benötigt? (*Ausführungszeit und Ressourcen-Überwachung*)
- Wie sieht die Struktur des Codes aus? (*Programmstruktur*)

Die folgende Tabelle zeigt, welche Art von Analyse für die einzelnen Informationen am aussagekräftigsten ist.

Tabelle 3.1:
Einsatzmöglichkeiten
eines Profilers

Zweck	Erfaßte Daten
Algorithmus testen	zeilenweise Information dynamische Aufrufhierarchie
Programmtest und Verifizierung	Ausführungszähler auf Routinen-Ebene dynamische Aufrufhierarchie
Überwachung der Ausführungszeit und Ressourcen	Ausführungszeit Anzahl der Ausführungen Interrupts Dateizugriffe Overlay-Zugriffe
Analyse der Programmstruktur	dynamische Aufrufhierarchie Dateizugriffe Analyse von Zeit und Anzahl der Ausführungen Interrupts Overlays

Testen von Algorithmen

Wenn Sie einen Algorithmus analysieren, konzentrieren Sie sich wahrscheinlich auf eine relativ kleine Anzahl an Routinen, so daß die Zeileninformation wichtiger ist als die Ausführungszeit. Gehen Sie wie folgt vor:

1. Isolieren Sie den Algorithmus und die zugehörigen Hilfsroutinen, indem Sie diese als Bereiche markieren.
2. Stellen Sie sicher, daß Bereichsmarkierungen auf alle Zeilen der Routinen gesetzt sind, mit denen der fragliche Algorithmus implementiert ist.

Die Beispiele in Kapitel 1 zeigen eine Algorithmus-Analyse, insbesondere in Beziehung zur Ausführungszeit.

Verifizierung und Testen

Für die Verifizierung und den Test von Programmen sind Informationen über die Abarbeitung von Codeblöcken wichtiger als Ausführungszeiten. Da bei der Verifizierung und dem Testablauf das Programm jedoch als Ganzes betrachtet wird, wollen Sie sehen, wie die Bestandteile in einem integrierten System zusammenarbeiten.

Da Sie es beim Testen und Prüfen mit großen Code-Abschnitten zu tun haben, brauchen Sie nicht so viele Einzelheiten wie bei der Analyse von Algorithmen berücksichtigen. Es ist jedoch sehr nützlich zu wissen, wie oft eine Routine aufgerufen wurde. Falls Sie den Test außerdem auf Gruppen von Routinen ausdehnen wollen, die eine Hierarchie bilden, kann die Kenntnis über die Anzahl der Ausführungen sehr nützlich sein, um

schnell überprüfen zu können, ob jede Verzweigung in einer **switch**-Anweisung oder bedingten Verzweigung mindestens einmal ausgeführt wurde.

Die Analyse eines Programms während der Standardtests kann bestimmte Programmbereiche aufzeigen, die nur selten oder gar nicht ausgeführt werden. Dies überprüfen Sie am besten im Coverage-Modus. Indem Sie zum Beispiel die Aufrufinformationen (*Call Paths*) im *Callers*-Fenster untersuchen und ein Quellcode-Listing (zusammen mit der Anzahl und/oder Ausführungszeit) über das *Module*-Fenster ausgeben, können Sie überprüfen, ob auch wirklich jede Anweisung in Ihrem Quellcode ausgeführt wird.

Ausführungszeit und Ressourcen- Überwachung

Zur Lokalisierung zeitraubender Programmstellen eines großen Programms benötigen Sie nur selten Informationen auf Zeilenebene. Um die Ausführungszeit zu ermitteln, müssen Sie wissen:

1. wieviel Zeit in den einzelnen Routinen zugebracht wird;
2. welche Zeiten von den Routinen der unteren Ebene in die Routinen der höheren Ebene fortgeschrieben werden.

Bevor Sie die Ausführungszeit eines Programms messen können, müssen Sie für alle Routinen einen Bereich markieren. In sehr großen Programmen sollten Sie die Angabe von Bereichsmarkierungen auf ein Modul beschränken.

Sobald die Bereichsmarkierungen in einem Modul gesetzt wurden, beginnt die Analyse mit einer nach und nach erfolgenden Gruppierung und Verfeinerung. Folgende Methoden dienen zu Verfeinerung der Analyse:

- Klammern Sie überflüssige Informationen mit Hilfe des Befehls *Filter* im lokalen Menü des Fensters *Execution Profile* aus.
- Entfernen Sie die Markierung von den Routinen, die Sie nicht messen wollen (mit Hilfe des Befehl *Remove* in den Fenstern *Module*, *Execution Profile* oder *Areas*).
- Kombinieren Sie die Zeiterfassung für die angegebenen Routinen (mit der Option *Timer*, die Sie entweder mit dem Befehl *Statistics / Profiling Options* oder dem lokalen *Options*-Befehl in *Areas* einstellen können).

Wenn Sie mit dem Programm, das Sie analysieren, noch nicht sehr vertraut sind, ist ein Vergleich der Ausführungszeit und des Leistungsverhaltens eines Programms mit dem Quellcode sehr nützlich.

Untersuchung der Programmstruktur

Eine der besten Methoden zur Untersuchung eines Ihnen unbekannten Codes ist die Auswertung der dynamischen Aufrufhierarchie, die Turbo Profiler im Fenster *Callers* erzeugt. Diese Darstellung des Programm-

ablaufs zeigt die Hierarchie in der Programmstruktur. Obwohl Sie im *Callers*-Fenster jeweils nur die *Call Paths* von einer Routine sehen, können Sie alle aufgezeichneten Aufrufpfade mit dem Befehl *Print / Statistics* aus dem geöffneten *Callers*-Fenster heraus ausgeben.

Durch die Ermittlung der im Programm aufgerufenen Routinen, deren aufrufenden Routinen und Anzahl der entsprechenden Programmverzweigungen in den Aufrufpfaden, erkennen Sie, auf welche Routinen es ankommt. Sie können auch vorhersagen, welche Routinen auf einer höheren Stufe von den Änderungen auf einer niedrigeren Stufe beeinflusst werden.

Indem Sie die Zeit und Anzahl der Ausführungen in Betracht ziehen, bekommen Sie ein Gefühl dafür, welche Routinen, am wichtigsten sind. Die Überwachung der Datei- und Overlay-Zugriffe zeigt, welche Dateien während der Programmausführung zwischenzeitlich geöffnet und geschlossen und welche Overlays in den Speicher geladen werden. Solche Informationen sind durch eine Untersuchung des Quellcodes viel schwerer zu ermitteln.

Die Verbindung zwischen den Fenstern *Execution Profile*, *Module* und *Areas* ermöglicht Ihnen die schnelle Suche nach den angegebenen Symbolen und deren Daten. Damit wird die Verbindung zwischen funktionell verwandten, aber physikalisch getrennt vorliegenden Teilen des Quellcodes geschlossen.

Welcher Analyse-Modus soll verwendet werden?

Eine wichtige Überlegung für die Analyse betrifft die Entscheidung zwischen aktiver oder passiver Analyse. Sie können den Untersuchungsmodus unter *Profile Mode* im Dialogfenster *Profiling Options* festlegen (wählen Sie dazu *Statistics / Profiling Options*).

Wenn Sie sich schließlich davon überzeugen konnten, daß das Programm korrekt läuft, sind die aktive und die passive Analyse wichtige Werkzeuge für die Verbesserung des allgemeinen Leistungsverhaltens des Programms. Andererseits stellt das Coverage-Profiling ein wichtiges Hilfsmittel bei der Entwicklung des Programmcodes dar.

Ruft das Programm auch wirklich alle Routinen zur richtigen Zeit auf? Gibt es Code-Bereiche, die nie ausgeführt werden? Das sind wichtige Fragen, die bei der Analyse im Coverage-Modus geklärt werden.

Aktive Analyse

Der Standard-Modus von Turbo Profiler ist die aktive Analyse. Damit werden Zeit und Anzahl der Ausführungen wie auch andere Daten (zum Beispiel Aufruf-Protokolle oder DOS-Interrupts), deren Erfassung im *Statistics*-Menü aktiviert ist, automatisch erfaßt.

Wenn Sie ein Programm im aktiven Modus untersuchen, kommt es darauf an, wie oft Ihr Programm eine Bereichsmarkierung passiert. Sie können zum Beispiel jede Zeile in einem Programm mit Ausnahme einer Schleifenanweisung markieren. Wenn das Programm 95% der Gesamtzeit in dieser Schleife zubringt, wird das Programm dadurch kaum verlangsamt.

Im Abschnitt „Schnellere Analysen“ erfahren Sie, wie Sie Analysen schneller durchführen lassen können.

Der Profiler verlangsamt die Programmausführung nur dann, wenn bei der Ausführung einer Quellenweisung sehr viele Daten erfaßt werden müssen. Ist dies der Fall, können Sie auf die passive Analyse umschalten. Dann werden alle automatischen Aufrufe für die Datenerfassung abgeschaltet, die der Profiler in der Regel ausführt, wenn eine Bereichsmarkierung passiert wird.

Passive Analyse

Wenn Ihr Programm sehr langsam läuft und die Erfassung der Anzahl der Ausführungen sowie der Aufruf-Hierarchien weggelassen werden können, sollten Sie die passive Analyse verwenden. Im passiven Modus erfaßt Turbo Profiler nur die zeitabhängigen Statistiken für die Bereiche (wie die Ausführungszeit, die Interrupt-Aufrufe und die Dateizugriffe). Dadurch kann Ihr Programm vom Profiler sehr viel schneller ausgeführt werden.

In der passiven Analyse unterbricht Turbo Profiler die Ausführung Ihres Programms in regelmäßigen Abständen, um den Wert des Programmzählers, CS:IP, festzuhalten. Wenn der erfaßte Wert auf eine Adresse in einem beobachteten Bereich zeigt (ein markierter Bereich), erhöht der Profiler die Anzahl der Zeittakte für diesen Bereich. Wenn der Wert nicht auf eine Adresse in einem markierten Bereich zeigt (zum Beispiel während eines DOS- oder BIOS-Interrupts), wird dieser Zeittakt nicht berücksichtigt.

Es ist nicht leicht, die Ergebnisse einer passiven Analyse richtig zu deuten, wenn Ihr Programm nicht lang genug läuft oder die Statistik nicht über mehrere Programmläufe zusammengezählt wird. Einige Bereiche Ihres Codes werden möglicherweise überhaupt nicht angezeigt, obwohl sie ausgeführt werden, da sie nie in dem Moment ausgeführt werden, zu dem der Profiler die Programmausführung unterbricht. Wenn Sie ein Programm im passiven Modus untersuchen, erhalten Sie durch mehrere Programmläufe statistisch genauere Meßwerte (stellen Sie dazu *Run Count* im Dialogfenster *Profiling Options* auf eine entsprechende Anzahl von Durchläufen ein).

Wenn Sie auf passive Analyse umstellen, kommt es zu keiner merklichen Verlangsamung der Programmausführung. Sie erhalten aber unter Umständen nicht alle gewünschten Informationen. Zum Beispiel ist es nicht

möglich, die Anzahl der Aufrufe zu ermitteln oder die Aufrufe nachzuverfolgen. Die Interrupt-Aufrufe und Dateizugriffe lassen sich jedoch beobachten. (Die passive Analyse führt zu keiner wesentlichen Verlängerung der Laufzeit im Profiler, aber es muß auf einige Details in den Ergebnissen verzichtet werden.)

Was Sie immer im Auge behalten sollten

Einige der Daten, die Sie in der passiven Analyse erfassen, führen Sie vielleicht auf eine falsche Fährte, wenn Sie die folgenden Gesichtspunkte bei der Auswertung Ihrer Ergebnisse nicht berücksichtigen:

- Wenn Ihr Programm viele Dateizugriffe macht, berücksichtigt der Profiler die Zeit für diese Dateizugriffe der aufrufenden Routinen nur im aktiven Modus, nicht aber im passiven Modus.
- Wenn Ihr Programm einen Interrupt aufruft, der nicht als Bereich markiert ist, zählt der Profiler die Zeitdauer nur im aktiven Modus zur aufrufenden Routine hinzu, während die Zeit für den Interrupt im passiven Modus nicht berücksichtigt wird.

Coverage-Analyse

Bei der Coverage-Analyse wird die Programmstruktur und die Reihenfolge bei der Ausführung der einzelnen Programnteile überprüft. Da die Coverage-Analyse die Ausführung des Programms nicht verlangsamt, werden bei einem Analysedurchgang gleich zwei Fliegen mit einer Klappe geschlagen: Es wird ermittelt, wie oft ein Codeblock ausgeführt wird (die maximale Zahl an Durchläufen wird gesetzt mit *Statistics / Profiling Options*); damit wird einerseits nachgewiesen, daß ein bestimmter Block tatsächlich aufgerufen wird und andererseits festgestellt, ob er zu selten oder zu oft aufgerufen wird.

Schnellere Analysen

Jedesmal, wenn Ihr Programm in eine Routine kommt, in der ein Bereich für die Erfassung von Daten definiert ist, muß Turbo Profiler eine Bearbeitung vornehmen (die *Buchführung*). Die Ausführungszeit eines Programms unter der Programmsteuerung von Turbo Profiler hängt davon ab, wie oft Bereichsmarkierungen überschritten werden und auf welche Art und Weise die Informationen für die am häufigsten angetroffenen Bereichsmarkierungen erfaßt werden. Je größer die Informationsflut ist, die aufgezeichnet wird (insbesondere die Aufrufhierarchie), desto länger dauert die »Buchhaltung« in diesem Bereich.

Hin und wieder kann es sein, daß die Ausführung Ihres Programms unter Turbo Profiler zu langsam ist. Das kann daran liegen, daß Ihr Programm sehr oft eine tief verschachtelte Routine aufruft und die Option für den Aufruf-Stack unter *All Areas* auf *All* eingestellt ist. Wenn Sie eine dieser tief verschachtelten Routinen als einen Bereich definiert haben, verbringt Turbo Profiler sehr viel Zeit damit, die Aufrufe festzuhalten. (Auch wenn

Ihr Programm langsamer läuft, werden die zeitlichen Informationen von Turbo Profiler korrekt erfaßt.)

Um festzustellen, ob Ihr Programm sehr häufig eine tief verschachtelte Routine aufruft, wechseln Sie in das Fenster *Execution Profile* und lassen sich die Bereiche und die Anzahl ihrer Ausführungen anzeigen. (Setzen Sie dazu die Option *Display* des lokalen Menüs auf *Counts*.) Damit erhalten Sie ein Histogramm über die Anzahl der Ausführungen, das nach steigender Anzahl der einzelnen Bereiche sortiert ist.

Wenn das Programm eine oder mehrere Routinen sehr viel öfter aufruft als die Restlichen, können Sie erstere aus der Liste der angezeigten Bereiche mit dem Befehl *Filter / Command* des Fensters *Execution Profile* ausschließen. Sie können die Bereichsmarkierungen mit dem lokalen Befehl *Remove* in den Fenstern *Module*, *Areas* und *Execution Profile* auch gleich ganz löschen.

Verbesserung der statistischen Genauigkeit

Wenn Sie nicht genug Daten erfassen (da ihr Programm vielleicht zu schnell läuft, um eine statistisch signifikante Anzahl Daten erfassen zu können) oder wenn Sie statistisch gesehen stark ungleichmäßig verteilte Daten erhalten (aufgrund eines Resonanz-Effekts: die Frequenz des Zeittakts fällt mit der Ausführungsfrequenz eines Teils Ihres Programms zusammen), sind Sie nicht in der Lage, die richtigen Entscheidungen für die Veränderungen an Ihrem Quellcode zu treffen. Nachfolgend wird beschrieben, was man in solch einem Fall tun kann.

Zuwenig Daten

Um die Genauigkeit der Zeitstatistik zu verbessern und einen statistisch besser abgesicherten Durchschnitt zu erhalten, starten Sie Ihr Programm mehrmals, indem Sie die Option *Run Count* im Dialogfenster *Profiling Options* entsprechend einstellen. Wird Ihr Programm beendet und daraufhin ein weiteres Mal gestartet, addiert Turbo Profiler die Zeiten für einen erneuten Durchlauf zu den akkumulierten Zeiten der früheren Programmläufe. Dies wird solange fortgesetzt, bis die in der Option *Run Count* eingestellte Zahl an Programmausführungen erreicht ist.

Vorsicht!

Wenn Sie Ihr Programm nach Vollendung der vorgegebenen Anzahl an Durchläufen ein weiteres mal starten, werden alle statistischen Informationen gelöscht, um eine weitere Anzahl an Testläufen vorzubereiten. Sichern Sie die statistischen Ergebnisse (oder werten Sie sie aus), bevor Sie ein weiteres mal eine vorgegebene Anzahl an Testläufen auslösen.

Resonanzen

Resonanzprobleme können zum Beispiel auftreten, wenn die Frequenz eines Schleifendurchlaufs zufällig mit der Frequenz übereinstimmt, die der Zeitgeber-Baustein liefert. Wenn der Profiler aufgrund von Resonanzeffekten falsche Daten liefert, verwenden Sie die Einstellung *Clock speed* im

Dialogfenster *Profiling Options*, um den Zeittakt des Profilers auf einen anderen Wert (zwischen 18 und 1000 Takten pro Sekunde) einzustellen. Wählen Sie eine Frequenz, die kein Vielfaches des Zeittakts darstellt, durch den die Resonanz erzeugt wurde. Wenn Ihr Programm zum Beispiel bei 100 Zeittakten pro Sekunde einen Resonanz-Effekt zeigt, sollten Sie es einmal mit einer Frequenz von 130 oder 70 Zeittakten versuchen.

Wenn Sie meinen, daß Resonanz die Statistik verfälscht hat, probieren Sie einige andere Taktfrequenzen, die keine ganzzahligen Vielfache der Ausgangswerte darstellen (mit Sicherheit sind dies Primzahlen), und vergleichen die erhaltenen Meßdaten. Wenn Resonanzen das Problem darstellen, sollten sich die einzelnen Statistiken deutlich unterscheiden.

Die Taktfrequenz kann nur im passiven Modus verändert werden; im aktiven Modus wird der Takt des Zeitgebers nicht verwendet.

Je schneller der Zeittakt, desto genauer kann Turbo Profiler feststellen, wo Ihr Programm die Zeit zubringt.

Sollte man also nicht mit einer Frequenz von 1000 Zeittakten pro Sekunde die genauesten Meßdaten erhalten? Nicht unbedingt. Je schneller der Zeittakt, desto langsamer wird Ihr Programm ausgeführt (da Turbo Profiler bei jedem Zeittakt gewisse Prüfoperationen durchführen muß). Wenn Sie also eine höhere Genauigkeit wollen als mit 100 Zeittakten pro Sekunde, sollten Sie die Frequenz des Zeitgebers erhöhen, bis Sie einen für Sie akzeptablen Mittelweg zwischen Genauigkeit und Ausführungszeit gefunden haben.

Beachten Sie auch, daß, wenn eine Routine Ihres Programms die Interrupts sperrt, nur der erste Zeittakt während der Ausführung der Routine erfaßt wird und Ihre interne Systemuhr langsamer laufen wird. Vermeiden Sie eine Taktfrequenz, die mehrere Takte erzeugt, während die Interrupts gesperrt sind.

Einige Hinweise zur Analyse von Overlays

Mit Hilfe von Overlays können große Programme auch bei begrenzten Speicherverhältnissen ausgeführt werden, indem Teile des Programms auf ein Laufwerk ausgelagert werden und nur der benötigte Code geladen wird. Wenn Sie Overlays verwenden, teilen sich mehrere verschiedene Module denselben Speicherbereich – so daß der gesamte RAM-Bedarf reduziert werden kann.

Leider kann der Austausch der Programmteile zwischen Laufwerk und Speicher zu einer erheblichen Verlangsamung des Programms führen, da der Zugriff auf Laufwerke sehr lang dauern kann. Weil auch das schnellste Laufwerk immer noch sehr viel langsamer ist als der RAM-Speicher, kann eine falsche Speicherverwaltung zu einer erheblichen Verschlechterung des Laufzeitverhaltens führen. Die Sache kommt noch schlimmer, da der Code der Overlay-Verwaltung im compilierten Programm norma-

lerweise nicht sichtbar ist. Turbo Profiler bringt den Code der Overlay-Verwaltung jedoch ans Licht, damit Sie das Programmverhalten Ihrer Overlays optimal einstellen können.

Um das Overlay-Verhalten genau einzustellen, müssen Sie die optimale Größe für den Overlay-Puffer ermitteln, den besten Algorithmus für die Verwaltung der Overlays im Puffer auswählen und einige Parameter einstellen, mit denen Sie dafür sorgen können, daß die am häufigsten verwendeten Overlays länger im Speicher bleiben.

Zu den im Fenster *Overlay* angezeigten Statistiken gehören:

- wie oft Ihr Programm ein Overlay lädt;
- eine zeitlich sortierte Folge der Ereignisse, in denen Ihr Programm Overlays lädt.

Die Information über die Anzahl der Ladevorgänge und die Ausführungsdauer eignet sich gut, um zu ermitteln, welche Overlays im RAM gehalten werden sollen. Durch einen Vergleich dieser Daten mit einer Messung der Routinen, in denen die Routinen nicht als Overlay vorliegen, können Sie entscheiden, welche Module zu Overlays gemacht werden sollten und welche nicht.

Mit der Aufzeichnung der Ereignisse können Sie die optimalen Algorithmen für die Overlay-Verwaltung auswählen. Nach einer Untersuchung der Overlay-Liste können Sie anhand der Zeitpunkte des Ladens entscheiden, welche der Module des Hauptprogramms sich besser als Overlays eignen würden und welche Overlays (bzw. dann natürlich Module) eher ins Hauptprogramm übernommen werden sollten.

Die Untersuchung objektorientierter Programme

Im allgemeinen unterscheidet sich die Analyse von objektorientierten Programmen kaum von der Analyse konventioneller Programme. Sie können solche Programme wie normale Programme behandeln und die Elementfunktionen oder Methoden wie normale Routinen behandeln.

Auswertung und Anwendung der Ergebnisse

Sie haben jetzt entschieden, welche Art von Statistik erfaßt werden soll, haben das Programm entsprechend angepaßt, und so oft aufgerufen, daß eine verlässliche Statistik erfaßt wurde (wenn Sie keinen groben Fehler gemacht haben). Was nun?

Jetzt folgt der interessante Teil der Aufgabe. Zunächst werten Sie die Daten aus, um herauszufinden, was die Daten des Profilers bedeuten. Dann wenden Sie Ihr neu erworbenes Wissen auf den Quellcode an, um Ihr Programm schneller und effizienter zu machen.

Wie werden die Profiler-Daten ausgewertet?

Die Fenster, die Sie in Turbo Profiler verwenden, lassen sich in zwei Kategorien einteilen: Es gibt *Quellcode*-Fenster und *Report*-Fenster.

Zu den Quellcode-Fenstern von Turbo Profiler zählen *Module*, *Areas*, *Routines* und *Disassembly (CPU)*. Bevor Sie eine Analyse starten, sollten Sie die Quellcode-Fenster verwenden, um die Bereichsmarkierungen und deren Analyse-Operationen festzulegen. Nachdem Sie die Analysedaten (in ein oder mehreren Report-Fenstern) untersucht haben, sollten Sie wieder zu den Quellcode-Fenster zurückkehren, um den Quellcode Ihres Programms auszuwerten.

Zu den Report-Fenstern von Turbo Profiler zählen die Fenster *Execution Profile*, *Callers*, *Overlays*, *Interrupts*, *Files* und *Coverage*. Sie können die Report-Fenster für die Darstellung der Meßdaten Ihres Programms verwenden, um die erfaßten Daten auszuwerten und festzustellen, durch welche Änderungen sich das Laufzeitverhalten des Programms verbessern läßt.

Das Fenster *Execution Profile*

Dieses Fenster steht im Mittelpunkt für die Verbesserung des Laufzeitverhaltens Ihres Programms. Im allgemeinen werden Sie zunächst die Anweisungen untersuchen, in denen das Programm am meisten Zeit zubringt. Anschließend suchen Sie sich die Zeilen (oder Routinen) heraus, die das höchste Verhältnis von Ausführungszeit zur Anzahl der Ausführungen aufweisen. Und abschließend wenden Sie sich den Routinen zu, die am stärksten zur „Ausführungszeit pro Aufruf“ beitragen.

Das Fenster *Callers*

Haben Sie eine Routine gefunden, die Sie verbessern wollen, dann verwenden Sie das *Callers*-Fenster, um alle Bereiche in Ihrem Programm zu ermitteln, von denen diese Routine aufgerufen wird. Das *Callers*-Fenster zeigt, wie oft die Routine aufgerufen wurde und von welchen Routinen aus die Aufrufe erfolgten.

Das Fenster *Overlays*

Mit Hilfe des Fensters *Overlays* können Sie häufige Overlay-Zugriffe von Routinen aufdecken, die besser nicht als Overlay realisiert werden.

Das Fenster *Interrupts*

Das *Interrupts*-Fenster zeigt alle der (ausgewählten) Interrupts, die durch Ihr Programm erfolgten. Dadurch werden Sie vielleicht veranlaßt, zum Beispiel die Video-Ausgabe mancher Zeilen zusammenzufassen oder die Ein-/ Ausgabe auf die Festplatte bzw. Diskette in Programmen mit vielen Dateizugriffen zu puffern.

Das Fenster *Files*

Im *Files*-Fenster werden die Anzahl aller Lese- und Schreibzugriffe auf Dateien in Ihrem Programm offenbart. In Anwendungen mit vielen Ein-/ Ausgaben können Sie sofort erkennen, welche Dateien Ihre Aufmerksamkeit verdienen.

Das Fenster *Coverage*

Das Fenster *Coverage* gibt in seiner Grundeinstellung Codeblöcke aus, die während des Analysedurchgangs nicht ausgeführt wurden. Damit können Sie Codebereiche aufspüren, die nicht aufgerufen werden (sog. tote Bereiche) und Bereiche, die eigentlich aufgerufen werden sollten. Darüber hinaus kann das Fenster *Coverage* so eingerichtet werden, daß es angibt, wie oft ein bestimmter Codeblock aufgerufen wurde, während das Programm lief.

Filtern der erfaßten Daten

Das Fenster *Execution Profile* enthält einige lokale Menüs, mit denen sich bestimmte Daten zwischenzeitlich oder permanent für die Anzeige herausfiltern lassen. Nachfolgende Tabelle faßt die Filter-Optionen von Turbo Profiler zusammen:

Tabelle 3.2:
Lokale Menübefehle
zum Filtern von
Statistiken

Fenster	Lokaler Menübefehl	Beschreibung
Execution Profile	<i>Filter</i>	Entfernt zwischenzeitlich die Meßdaten des aktuellen Bereichs oder zeigt nur die Statistik des aktuellen Moduls oder stellt wieder alle erfaßten Statistiken im Fenster dar. (Sie wählen <i>Current</i> , <i>Module</i> oder <i>All</i> im Menü <i>Filter</i> aus.)
	<i>Remove</i>	Löscht die aktuellen Bereichsstatistiken permanent von den erfaßten Daten. Verwenden Sie diese Option nur mit Vorsicht!
Files	<i>Collection</i> (oberer Ausschnitt)	Schaltet mit <i>Disable</i> die Erfassung ab.
	<i>Detail</i> (oberer Ausschnitt)	Mit <i>Disabled</i> werden nur die Dateizugriffe zum Öffnen und Schließen angezeigt. Mit <i>Enabled</i> werden auch die Lese- und Schreibzugriffe angezeigt.
	<i>Display</i> (unterer Ausschnitt)	Stellt jedes Ereignis entweder als Element einer Balkengrafik oder als Text mit Zeitpunkt und Zeitdauer des Ereignisses dar.
Interrupts	<i>Remove</i> (oberer Ausschnitt)	Entfernt den aktuell ausgewählten Interrupt aus dem obersten Ausschnitt.
	<i>Display</i> (unterer Ausschnitt)	Zeigt die Interrupt-Statistik entweder als zusammenfassendes Histogramm mit Zeiten, Aufrufen oder beidem an oder als eine detaillierte Folge von Ereignissen.
Overlays	<i>Display</i>	Zeigt die Statistik für die Overlay-Zugriffe entweder als Anzahl mit einer Zusammenfassung der Speicherbelegung oder als eine History-Liste mit einer detaillierten Abfolge der Ereignisse, die in einer Zeile mit den Zeiten für das geladene Overlay dargestellt werden.

Coverage	<i>Display</i> (rechter Ausschnitt)	Legt die Anzahl der Blöcke fest, die angezeigt werden: (1) Anzeige aller Blöcke (2) Beschränkung auf die Blöcke, die nicht ausgeführt wurden.
	<i>Sort</i>	Bestimmt die Reihenfolge, in der die Blöcke angezeigt werden (entweder nach Adresse oder der Anzahl der Ausführung).
	<i>Group</i>	Bestimmt die Anzeige von Blöcken entweder (1) alle Blöcke oder (2) Zusammenfassung der Blöcke nach Routinen oder (3) nach Modulen.

Wenn Sie die Option *Remove* im lokalen Menü von *Execution Profile* auswählen, um die Statistik eines Bereichs permanent herauszufiltern, geschieht folgendes:

- Der Report wird entsprechend der nicht mehr berücksichtigten Zeitdauer angepaßt.
- Der Prozentsatz der restlichen Bereiche wird neu berechnet:
(*neue Gesamtzeit = Gesamtzeit - Zeit im entfernten Bereich*)
- Die Markierung des Bereichs wird aus dem *Module*-Fenster entfernt.
- Der Bereich wird aus der Bereichsliste im Fenster *Areas* entfernt.

Hinweis

Wenn Sie aus dem Fenster *Execution Profile* einen Bereich entfernen, werden die Statistiken für die verbleibenden Bereiche nur für den aktuellen Durchlauf neu berechnet. Ein darauf folgender weiterer Durchlauf wird sehr wahrscheinlich die in dem entfernten Bereich verbrauchte Zeit wieder angeben, jedoch in ein oder mehrere andere Bereiche hinein verschoben. Um einen Bereich auf Dauer aus der Statistik zu entfernen (und auch den Eintrag für die Zeit, die darin verbraucht wurde), müssen Sie diesen Bereich auf *Disable* setzen. Sie dürfen dann nicht vergessen, den folgenden Bereich auf *Enable* zu setzen.

Überarbeiten Sie Ihr Programm

Nachfolgend finden Sie eine allgemeine Strategie, um das Auffinden von Routinen in Angriff nehmen zu können, in denen durch einfache Änderungen in den Kontrollstrukturen das Laufzeitverhalten Ihres Programms verbessert werden kann:

1. Suchen Sie nach großen Routinen, die einen unverhältnismäßigen Anteil an der Ausführungszeit beanspruchen, oder nach Routinen, die häufig aufgerufen werden. Ausgehend von der obersten Programmebene verfolgen Sie den Programmablauf durch die verschiedenen Aufrufebenen, in denen Sie nach Möglichkeiten Ausschau halten, unnötige Aufrufe und Operationen zu eliminieren.

2. Suchen Sie nach Anweisungen und Routinen, die ein hohes Verhältnis von Zeit zu Anzahl aufweisen. Setzen Sie dazu die Option *Display* im Fenster *Execution Profile* auf *Both* oder *Per Call*. Anschließend suchen Sie nach den Bereichen, die einen langen Balken für die Zeit und nur einen kurzen Balken für die Anzahl aufweisen. Anweisungen und Routinen dieser Art stellen oft einen ineffizienten Code-Abschnitt dar. Codieren Sie den Abschnitt neu, um das gewünschte Ergebnis effizienter zu erhalten.
3. Als letzter Bereich bleibt die Optimierung der internen Schleifen eines Programms. Einige Vorschläge:
 - Schleifen auflösen;
 - Resultate in Schleifen zwischenspeichern;
 - Berechnungen, deren Ergebnisse sich nicht ändern, außerhalb von Schleifen durchführen;
 - Codierung in Assembler.

Normalerweise werden Sie durch die Optimierung in den Schleifen, insbesondere im Vergleich zu Änderungen der Kontrollstrukturen, Algorithmen oder Datenstrukturen, nur geringe Verbesserungen feststellen.

Neben diesen allgemeinen Hinweisen, gibt es noch einige andere Möglichkeiten, wie Sie Ihr Programm verbessern können:

- Ändern Sie die Datenstrukturen und Algorithmen;
- Speichern Sie vorausberechnete Ergebnisse;
- Speichern Sie Daten, auf die häufig zugegriffen wird;
- Werten Sie nur die Daten aus, die unbedingt erforderlich sind;
- Optimieren Sie Schleifen, Prozeduren und Ausdrücke.

Ändern der Datenstrukturen

Verwenden Sie bessere Datenstrukturen oder Algorithmen. Eine Quicksort-Routine ist im allgemeinen sehr viel schneller als ein Bubble-Sort für eine zufällige Verteilung an Vergleichsdaten. Schlagen Sie in einem Buch über Datenstrukturen und Algorithmen weitere Beispiele nach.

Stellen Sie zeitkritische Berechnungen von Real- auf Integer-Zahlen um, wie zum Beispiel bei der Verwaltung von Fenstern oder Strings für eine schnelle Bildschirmausgabe und Grafik-Routinen. Verwenden Sie Long-Integer-Daten für Datenumwandlungen bzw. für alle Werte, die keine Fließkommazahlen erfordern.

Anstatt ein Array von Textzeilen zu sortieren, sollten Sie ein Array von Zeigern auf die Textzeilen sortieren. Alle Zugriffe auf Strings erfolgen über Zeiger. Um einen Text zu sortieren oder eine neue Zeile einzufügen, brauchen Sie nur die Reihenfolge der Zeiger umzustellen und nicht gleich die ganzen Zeilen im Text zu vertauschen.

**Speichern von vor-
ausberechneten
Ergebnissen**

Legen Sie zum Beispiel eine Tabelle mit vorausberechneten Sinus-Werten an, in der Sie den Sinus als Funktion des Winkels über einen Integer-Index berechnen können.

**Häufig verwendete
Daten zwischen-
speichern**

In C wird die systemnahe Ein-/Ausgabe in Dateien gepuffert. Die Routine **getc** liest beim ersten Zugriff einen ganzen Sektor von der Disk in einen Puffer, liefert aber nur das erste eingelesene Zeichen zurück. Der nächste Aufruf von **getc** liefert das nächste Zeichen im Puffer, usw., bis der Puffer ausgelesen ist. Dann liest **getc** den nächsten Sektor von der Disk.

In einem interaktiven Editor oder einer Dump-Routine für Dateien können zum Beispiel eine Reihe von Puffern vorkommen, die im Rahmen einer *garbage collection* auf dem letzten Stand gehalten werden, wenn das Programm auf Benutzereingaben wartet. Sie haben vielleicht sogar zwei Puffer, die immer eine Bildschirmseite mit den Informationen vom Anfang oder Ende der Datei enthalten. In zwei weiteren Puffern könnten die Bytes des vorigen oder nächsten Bildschirms der Datei relativ zur aktuellen Position auf dem Bildschirm stehen. Auf diese Weise kann Ihr Programm unter Berücksichtigung der häufigsten Positionierbefehle den Bildschirm anpassen, ohne auf das Laufwerk zugreifen zu müssen.

**Nur erforderliche
Daten auswerten**

Ordnen Sie Abfragen und Verzweigungen so an, daß die wahrscheinlichsten Ergebnisse als erste ausgewertet werden können; die weniger wahrscheinlichen müssen dann nicht so oft untersucht werden.

In einer großen Tabelle zum Nachschlagen werten Sie nur die Einträge aus, die Sie benötigen, und verwenden Sie ein zusätzliches Array, um die bereits berechneten Einträge zu verfolgen.

Sie brauchen die Länge einer Zeile nur dann berechnen, wenn Sie die Ausgabe neu formatieren müssen – und nicht jedesmal, wenn eine Zeile aus einer Datei eingelesen wird.

**Optimierung beste-
hender Codes**

Schleifen, Prozeduren und Ausdrücke bieten oft die Gelegenheit für Verbesserungen:

Schleifen

- Falls möglich, sollten Berechnungen immer aus Schleifen herausgenommen werden. Immer wieder denselben Wert in einer Schleife zu berechnen, ist zum einen sehr zeitaufwendig und zum anderen nicht nötig. Die Ergebnisse aufwendiger Berechnungen sollten zwischengespeichert werden.

Eine Sortier- und Einfüge-Routine muß zum Beispiel bei der Verarbeitung eines Arrays nicht jedes Zahlenpaar austauschen. Wenn Sie den

Wert des Anfangselements speichern, braucht man in der inneren Schleife nur noch das nachfolgende Element verschieben, wenn das Element kleiner als das erste Element ist. Wenn die Prüfung scheitert, fügen Sie den gespeicherten Wert an der aktuelle Position ein. Diese Vorgehensweise ersetzt die aufwendige Austausch-Operation für die einzelnen Elemente in einem traditionellen Sortier-Algorithmus, der auf Einfügen basiert.

- Wenn zwei Schleifen mit denselben Daten eine ähnliche Operation durchführen, sollten Sie die beiden Schleifen zu einer Schleife kombinieren.
- Reduzieren Sie zwei oder mehr Prüfungen von Bedingungen in einer Schleife zu einem Test, falls dies möglich ist.

Addieren Sie zum Beispiel ein zusätzliches Element zum Array und initialisieren Sie dieses mit einem leicht identifizierbaren Wert, der den Abbruch der Schleife veranlaßt. (Dies ist auch die Methode, wie Strings in C verarbeitet werden.)

- Auflösung von Schleifen.

Zum Beispiel sollten Sie

```
for (x = 0; x < 4; x++)
    y += elemente[x];
```

ersetzen durch:

```
elemente[0] + elemente[1] += elemente[2] += elemente[3];
```

Routinen

- Schreiben Sie Routinen, die sehr häufig aufgerufen werden, als inline-Routinen oder ersetzen Sie deren Definitionen durch inline-Makros.
- Verwenden Sie Co-Routinen für Algorithmen, in denen große Datenmengen mehrmals durchgegangen werden (siehe **setjmp** und **longjmp** in C).
- Lösen Sie Ihre rekursiven Routinen mit Hilfe eines explizit verwalteten Daten-Stacks auf.

Ausdrücke

- Verwenden Sie Initialisierungen, die zum Zeitpunkt der Compilierung vorgenommen werden können.
- Kombinieren Sie zurückgelieferte Ergebnisse in einem einzigen Aufruf.
- Schreiben Sie zum Beispiel Funktionen, in denen Sinus/Kosinus, Quotient und Rest oder X- und Y-Koordinaten als Datenpaar zurückgegeben werden.
- Ersetzen Sie den Index-Zugriff auf Arrays durch Zeiger-Adressierung.

Zusammenfassung

In diesem Kapitel wurden die meisten Gesichtspunkte behandelt, die vor, während und nach einer Analyse zu beachten sind. Wir haben gezeigt, wie Sie Ihr Programm und sich selbst auf die Analyse vorbereiten müssen. Sie erhielten Hinweise und Ratschläge für die Durchführung der Analyse. Und wir haben Ihnen einige Vorschläge gemacht, wie Sie die Ergebnisse nach der Analyse verwenden können. Im nächsten Kapitel erfahren Sie Näheres über die Funktionsweise von Turbo Profiler.

Die Funktionsweise von Turbo Profiler

Falls Sie Turbo Profiler optimal ausnützen wollen, sollten Sie die internen Funktionsabläufe verstehen. Wenn Sie wissen, was der Profiler jedesmal macht, wenn eine Bereichsmarkierung erreicht wird, oder was passiert, wenn die Programmausführung vom Profiler unterbrochen wird, sind Sie in der Lage, Ihre Methoden zur Gewinnung und Auswertung von Informationen entsprechend zu verfeinern.

Betrachten wir dazu zunächst den Quellcode von PTOLL:

```
#include <stdio.h>
#include <dos.h>

void main()
{
    printf("Eintritt in main\n");
    route66();
    printf("Zurück in main\n");
    delay(1000);
    highway80();
    printf("Zurück in main\n");
    delay(1000);
    printf("Verlassen von main\n\n");
}

route66()
{
    printf("Eintritt in Route66\n");
    delay(2000);
    printf("Verlassen von Route66\n");
}
```

```
highway80()  
{  
    printf("Eintritt in Highway80\n");  
    delay(2000);  
    printf("Verlassen von Highway80\n");  
}
```

Setzt man auf alle Routinen (*All Routines*) einen Bereich, so erhält man insgesamt eine vierfache Unterteilung für die Erfassung der Zeit und der Anzahl der Ausführungen.

Unsichtbare Mautstationen

In diesem Abschnitt verfolgen Sie die Programmausführung, um zu sehen, was beim Durchlaufen einer Bereichsmarkierung passiert. Sie können sich diesen Vorgang wie das Passieren einer Mautstation vorstellen. Wenn Sie eine Mautstation durchlaufen, befinden Sie sich auf einem Straßenabschnitt, der zu dieser Mautstation gehört, bis Sie zur nächsten Mautstation kommen.

Bevor Sie die erste Mautstation passieren und nachdem Sie die letzte Mautstation hinter sich gelassen haben, befinden Sie sich in einem Freiraum, der nicht überwacht wird. Ab dem Passieren einer Mautstation ist jedoch bekannt, wie lange Sie auf dieser Straße gefahren sind. Außerdem wird festgehalten, wie oft Sie an einer Mautstation vorbeikommen. Das einzige Problem an diesem Vergleich mit einer Autobahn ist die Tatsache, daß Sie auf einer richtigen Autobahn immer nur in eine Richtung fahren können: Schleifen und Sprünge erfordern einen Hubschraubertransport, mit dem Sie an einen zurückliegenden Abschnitt auf der Straße zurückkehren können.

Erfassen von Zeit- und Zählerwerten

Bevor Sie in den Block des Hauptprogramms gelangen, wird der C-Startup Code ausgeführt. Hier ist noch Niemandsland. Die Zeit, die hier verstreicht, wird nicht erfaßt werden, außer Sie haben im Startup-Code explizit einen Bereich angegeben.

Sobald Sie zur Bereichsmarkierung (Mautstation) in **main** kommen, wird der Zähler für **main** um eins erhöht. Jeder Zeittakt zwischen dem Zeitpunkt des Eintritts in **main** und des Aufrufs von **route66** wird dem Konto von **main** hinzugerechnet.

Als nächstes ruft **main** die Routine **route66** auf, mit der Sie auf einen neuen Autobahnabschnitt kommen. Zu dem Zeitpunkt, da die Programmausführung die Bereichsmarkierung (Mautstation) erreicht, werden mehrere Vorgänge ausgelöst:

- Der aktuelle Bereich wird auf **route66** gesetzt.
- Die Statistiken für den aufrufenden Programmteil (in diesem Fall **main**) wird auf einen Stack gelegt.
- Der Zähler von **route66** wird um eins erhöht.

Jeder Zeittakt, der von jetzt bis zur Rückkehr von **route66** abläuft, wird automatisch in der Zeitstatistik von **route66** erfaßt. Die Zeiterfassung für das gesamte Programm wird ebenfalls bei jedem einzelnen Zeittakt erhöht.

Sobald die Ausführung des Programms von **route66** zurückkehrt, holt der Profiler die Daten für den aufrufenden Programmteil vom Stack. Jeder Zeittakt, der ab jetzt bis zum Aufruf von **highway80** vorkommt, wird wieder in den Zählzeiten von **main** und der Gesamtstatistik berücksichtigt. (Falls Sie dies nachprüfen wollen, schalten Sie die Bereichsmarkierung für **route66** aus und vergleichen das Ergebnis mit einer Analyse, in der die Bereichsmarkierung noch gesetzt war. Es sollte sich ungefähr dieselbe Zeitdauer für die Ausführung des gesamten Programms ergeben. Die Ausführungsdauer von **main** wird sich jedoch um die Größe von **route66** erhöhen.)

Der Aufwand für einen Routinen-Aufruf

Es ist vielleicht ganz interessant festzustellen, wieviel Zeit beim *Aufrufen* einer Routine (zum Beispiel **route66**) verbraucht wird, wobei die Zeit *innerhalb* der Routine nicht berücksichtigt wird. Am einfachsten ist es, wenn man die Erfassung der Informationen beim Eintritt von **route66** ausschaltet und bei der Rückkehr von **route66** wieder einschaltet. Sie können diese Art von Information auch mit Hilfe einer *passiven Analyse* ermitteln, die in Kapitel 3 besprochen wird.

Bewegen Sie den Cursor also zur ersten Zeile von **route66** und wählen Sie die Option *Operation* im lokalen Menü des Fensters *Module*, um das Dialogfenster *Area Options* zu öffnen. Dort setzen Sie *Operation* (Erfassung) auf *Disable* (abgeschaltet).

Nachdem Sie die Erfassung der Daten beim Eintritt in **route66** abgeschaltet haben, wird die Erfassung bei der Rückkehr nicht wieder automatisch eingeschaltet. Sie müssen eine Bereichsmarkierung auf die abschließende geschweifte Klammer von **route66** setzen und die Erfassung dieses Bereiches mit *Enable* explizit wieder einschalten (im Dialogfenster *Area Operation*).

Was kostet eine Schleife?

Der Vergleich mit einer Mautstation erleichtert die Erklärung, warum sich beim Passieren einer Bereichsmarkierung und beim Rücksprung an eine Adresse, die vor dieser Markierung steht (durch eine Schleife oder eine Sprunganweisung) der aktuelle Bereich nicht ändert. Auch wenn Sie sich außerhalb des Gültigkeitsbereichs der Markierung befinden, haben Sie

noch keine neue Markierung passiert. Jeder Zeittakt wird immer der zuletzt passiertten Markierung zugeordnet.

Mit diesem Wissen schauen Sie sich das nächste Programm an:

```
#include <stdio.h>
#include <dos.h>
verfahren() ();

=> void main()
{
    printf("Eintritt in main\n");
    verfahren();
    delay(1000);
    printf("Verlassen von main\n\n");
    delay(1000);
}

=> verfahren()
{
    int i;
    printf("Suche nach der Straße...\n");
    delay(100);
    for (i=0; i<10; i++)
    {
        printf("Frage nach dem Weg\n");
=>        printf("Falsche Richtung\n\n");
        delay(1000);
    }
    printf("Wieder auf der Strecke\n");
}
```

In diesem Programm wird die Routine **verfahren** durch eine zusammengesetzte Anweisung in einer Schleife etwas unübersichtlicher. Nehmen wir einmal an, daß drei Markierungen gesetzt wurden: eine für **main**, eine für **verfahren** und eine Zeilenmarkierung für die Anweisung, in der die Meldung **Falsche Richtung** ausgegeben wird.

Die Situation wird etwas kompliziert, wenn die Routine **verfahren** aufgerufen wird. Beim ersten Eintritt in die Routine wird **verfahren** zum aktuellen Bereich. Die Zeit für den Ausdruck von **Suche nach der Straße** wird dieser Markierung zugeordnet.

Die Zeit für die Ausführung der Schleife ist immer noch mit der Routinen-Markierung verbunden. Wenn zum ersten Mal die **Frage nach dem Weg** auftaucht, bezieht sich die Zeit immer noch auf die Markierung für die Routine **verfahren**. Wird jedoch die Zeilenmarkierung von **Falsche Richtung** überschritten, so wird die restliche Zeit in der Routine dieser Zeilenmarkierung hinzugezählt.

Nur weil Sie in einen Bereich gelangen, der zuvor einer anderen Markierung zugeordnet war, bedeutet das noch lange nicht, daß sich der aktuelle Bereich ändert. Der aktuelle Bereich ändert sich nur dann, wenn eine Bereichsmarkierung überschritten wird. Dies kann zu unerwarteten Ergebnissen führen.

Wenn Sie zum Beispiel drei Markierungen für dieses Programm angeben (jeweils eine für das Hauptprogramm, **verfahren** und die Anweisung **Falsche Richtung**), dann werden etwa 84% der Programmzeit der Ausgabe von **Falsche Richtung** zugeordnet, während für **verfahren** nur 1 Prozent gemessen wird. Das liegt daran, daß neun der zehn Aufrufe von **Frage nach dem Weg** sowie alle nachfolgenden **delay**-Anweisungen in der Markierung des Bereiches **Falsche Richtung** addiert werden.

Falls Sie die Bereichsmarkierung für **Falsche Richtung** ausschalten, werden 84% der übrigen Ausführungszeit zur Routine **verfahren** protokolliert.

Betrachten Sie folgenden Programm-Abschnitt:

```
main()
{
    while ( !kbhit() )
    {
        Funktion1();
        Anweisung1;
        Anweisung2;
        Funktion2();
    }
}

Funktion1()
{
}

Funktion2()
{
}
```

Nehmen wir einmal an, daß für jede Routine im Modul ein Bereich festgelegt wird. Die Routinen **main**, **Funktion1** und **Funktion2** markieren jeweils den Anfang eines Bereiches. Beim Eintritt in **main** wird sofort die Bereichsmarkierung von **main** überschritten. Wenn das passiert, kommt Turbo Profiler intern an einen Breakpoint. Durch dieses Ereignis wird eine Variable gesetzt, in der festgehalten wird, daß **main** solange der aktuelle Bereich ist, bis der Profiler auf einen weiteren Breakpoint stößt. Gleichzeitig wird die Variable, in der die Anzahl der Ausführungen von **main** gespeichert wird, um eins erhöht.

Der Gültigkeitsbereich bezieht sich auf den Programmablauf und nicht auf die Struktur. Die Zuordnung erfolgt dynamisch und nicht statisch. Das bedeutet, daß **main** solange den aktuellen Bereich darstellt, bis **Funktion1** aufgerufen wird. Sobald die **Funktion1** aufgerufen wird, befindet sich der Profiler in einem neuen Bereich, bis ein weiterer Funktionsaufruf erfolgt oder die Ausführung von **Funktion1** abgeschlossen wird. Das heißt, daß der Profiler den aufrufenden Programmabschnitt (in diesem Fall **main**) auf einen Stack legt.

Wenn die **Funktion1** verlassen wird, passiert der Profiler eine Rücksprungmarkierung, die beim Eintritt in die **Funktion1** gesetzt wurde. Die Routine **main** wird wieder zum aktuellen Bereich. Alle Zeittakte während der Ausführung von **Anweisung1** oder **Anweisung2** werden der Zeit für den Bereich von **main** hinzugezählt.

Zwei Ereignisse gehen hier intern vor sich:

1. Jedesmal, wenn Sie in einen neuen Bereich kommen, ruft der Profiler eine interne Routine auf, die die Variablen entsprechend anpaßt und einen Stack mit den Aufrufen der Routinen verwaltet. Zu jedem Bereich gibt es zwei Variablen: die Anzahl der Ausführungen und die Dauer der Ausführung. Bei jedem Bereichseintritt, wird der Ausführungszähler für diesen Bereich inkrementiert.
2. Bei jedem Zeittakt ruft der Profiler eine Routine auf, in der ermittelt wird, in welchem Bereich sich das Programm gerade befindet, und inkrementiert dann die Zeittaktvariable dieses Bereiches um die entsprechende Zeitdauer.

Wenn das Programm beendet wird, wandelt Turbo Profiler die Anzahl der Zeittakte für jeden Bereich in eine Zeitdauer um (ausgehend von der Gesamtzahl an Zeittakten, die im gesamten Programm auftraten).

Routinen mit mehreren Ausgängen

Was ist zu tun, wenn eine Routine mehrere Ausgänge hat? Die Antwort hängt von der internen Behandlung eines Rücksprungs ab.

Hinweis

Auch wenn sich in Ihrer Routine unter Umständen mehrere explizite Rücksprungstellen befinden, wandeln C- und C++ Compiler von Borland alle Rücksprungstellen in Verzweigungen zu einem einzigen Austrittspunkt am Ende der Routine um. Die Zeile mit der Markierung für einen Rücksprung ist die abschließende geschweifte Klammer einer Routine. Diese entspricht der Rücksprunganweisung in Maschinencode, zu der alle anderen Rücksprunganweisungen im Quellcode der Routine verzweigen.

Die Erfassung von mehrfach aufgerufenen Routinen abschalten

Um die Erfassung einer Routine auszuschalten, müssen Sie eine Bereichsmarkierung mit der *Disable*-Option in der ersten Zeile der Funktion und einer *Enable*-Option in der Zeile hinter dem Aufruf der Funktion setzen. Wenn eine Funktion allderdings von verschiedenen Stellen aus aufgerufen wird, kann es schwierig werden, *Enable*-Optionen nach jedem Funktionsaufruf zu setzen. In diesem Fall wird die abschließende geschweifte Klammer der Funktion mit der *Enable*-Option markiert oder – noch besser – die Rücksprunganweisung des Funktionscodes.

Angenommen, es interessiert Sie nicht, wieviel Zeit das Programm in **Funktion1** zubringt. Dann setzen Sie in der ersten Zeile von **Funktion1** die *Disable*-Option. Bei der Rücksprunganweisung von **Funktion1** wird die Messung wieder aktiviert:

1. Gehen Sie in das Disassembly-Fenster (*View / Disassembly*).
2. Setzen Sie eine Bereichsmarkierung bei der *ret*-Anweisung der Funktion.
3. Gehen Sie zum Fenster *View / Areas*.
4. Setzen Sie die *Enable*-Option ([*Ctrl*] [*O*] [*E*]).

Jedesmal, wenn Sie von nun an **Funktion1** aufrufen, wird die Analyse beim Eintritt in die Funktion abgeschaltet und exakt an dem Punkt wieder aufgenommen, an dem sie verlassen wird.

Eine einfachere, wenn auch weniger genaue Methode zur Wiederaufnahme der Analyse besteht darin, die *Enable*-Option bei der schließenden Klammer der fraglichen Funktion zu setzen. Diese Methode ist etwas ungenauer, da auch die Zeit, die für den Rücksprung aus der Funktion benötigt wird, in die Erfassung der Zeitdauer eingeht.

Erfassung aufrufender Programmabschnitte

Eine aktive Routine ist eine Routine, die sich momentan auf dem Routinen-Aufrufstack des Profilers befindet. Bei einer aktiven Analyse (d.h., wenn der Profiler die Aufrufe und andere Daten protokolliert, die nichts mit der Zeitdauer zu tun haben) verwaltet Turbo Profiler einen eigenen Stack für die Aufrufe. Dieser Stack entspricht einem Stack, wie man ihn auch aus einem DOS-Programm kennt. Der Profiler-Stack ist jedoch vom Stack des Anwenderprogramms getrennt und wird nur dazu verwendet, um Informationen über die Aufrufe von Routinen zu speichern, die noch nicht beendet sind.

Für die Verwaltung eines Stack mit den aktiven Routinen verwendet Turbo Profiler zwei Arten von Bereichsmarkierungen:

- Bereichsmarkierungen für den Routineneintritt (Routinenmarkierungen)
- normale Bereichsmarkierungen (Zeilenmarkierungen)

Wenn der Profiler eine Routinenmarkierung erreicht, werden die Informationen der gerade aktiven Routine (der *letzten* Routinenmarkierung) auf den Stack gelegt. Die neue Routinenmarkierung wird dann zur aktiven Routinenmarkierung.

Wenn nun eine normale Bereichsmarkierung angetroffen wird, wirkt sich dies nicht auf die aktuelle Routine oder den Stack mit den aktiven Routinen aus. Die normale Markierung wird nur zum aktiven *Bereich*. Das bedeutet, daß der Profiler den zuvor aktiven Bereich vergißt. Die gerade aktive *Routine* bleibt jedoch auf dem Stack, bis der Profiler zu einer Rücksprungmarkierung kommt.

Wenn ein Rücksprung in einer aktiven Routine erfolgt, wird die Bereichsmarkierung, die mit dieser Routine verbunden ist, inaktiv. Die Routine an der Spitze des Profiler-Stack mit den aktiven Routinen wird vom Stack genommen und zur aktiven Routine, bis innerhalb dieser Routine eine Rücksprunganweisung kommt oder eine andere Routinenmarkierung passiert wird.

Der Profiler kann so die komplette Aufrufhierarchie für jede markierte Routine verwalten. Wenn Sie die Option *Statistics / Callers* für alle markierten Routinen eingeschaltet haben, wird jedesmal beim Überqueren einer Routinen-Markierung der gesamte Profiler-Stack in einen Puffer gespeichert, der direkt mit der Routinen-Markierung verbunden ist.

Wenn der Aufruf-Stack identisch mit einem früher bereits gespeicherten Aufruf-Stack ist, inkrementiert der Profiler einen Zähler, anstatt den Aufruf-Stack ein weiteres Mal zu speichern. Unterscheidet sich der Aufruf-Stack jedoch, dann legt der Profiler einen neuen Puffer an und protokolliert den Aufruf-Stack in diesem neuen Puffer. Dadurch ist es möglich, einen Datensatz für jeden Aufrufpfad zu einer Routine zu speichern und die Anzahl der Aufrufe der einzelnen Pfade festzuhalten.

Der Stack der aktiven Routinen wird in drei Menü-Optionen behandelt:

- *Statistics / Callers* (entweder ein-(*Enabled*) oder ausgeschaltet(*Disabled*));
- die Option *Callers* für jeden markierten Bereich im Fenster *Areas*;
- die Option *Stack* aus dem markierten Bereich im Fenster *Module*.

Über die lokalen Menüs der Fenster *Module* und *Areas* ist es möglich, eine feinere Einstellung für die Erfassung der Aufrufpfade vorzunehmen. Sie können die *Callers*-Option für jeden markierten Bereich separat einstellen. Sowohl der *Callers*-Befehl im lokalen Menü von *Module* als auch der *Options*-Befehl im lokalen Menü von *Areas* führen in ein Dialogfenster, in dem Sie *Callers* mit *All Callers*, *Immediate Caller* oder *None* angeben können.

- *All Callers* bedeutet, daß jedesmal beim Eintritt in die Routine der gesamte Aufruf-Stack mitprotokolliert wird.
- *Immediate Caller* bedeutet, daß nur der oberste Eintrag auf dem Aufruf-Stack protokolliert wird.
- *None* bedeutet, daß keine Stack-Informationen festgehalten werden, wenn das Programm in diese Routine eintritt.

Standardmäßig ist die Option *Callers* für alle Routinen-Markierungen auf *None* eingestellt.

Die Aktivierung von *Statistics / Callers* über das Hauptmenü hat dieselbe Wirkung, als hätten Sie im *Areas*-Fenster die *Callers*-Option auf *All Callers* für jede Bereichsmarkierung eingestellt. Sobald Sie jedoch irgendeine der *Callers*-Optionen von Hand eingestellt haben, ändert die Einstellung von *Statistics / Callers* auf *Enable* den Wert für die *Callers*-Optionen in diesen Bereichen nicht mehr.

Schaltet man dagegen die Option *Statistics / Callers* an dieser Stelle aus, protokolliert der Profiler keine Stack-Informationen mehr, ändert aber nicht die *Callers*-Einstellung im *Areas*-Fenster.

Stichproben oder Zählen

Dieser Abschnitt betrifft nur den passiven Modus.

Der Profiler mißt eigentlich gar nicht die Zeit: Er liefert eine sehr genaue Schätzung des Zeitbedarfs anhand der Anzahl an Zeittakten. Dies ist eine Form der *statistischen* Erfassung. Durch periodische Stichproben über den aktuellen Bereich und mit Hilfe eines entsprechenden Zählers – der jedes Mal erhöht wird, wenn der Bereich während eines Timer-Interrupts aktiv ist – kann eine Schätzung über die Zeitdauer gemacht werden, wie lange sich der Profiler in einem bestimmten Bereich aufgehalten hat.

Der Profiler kennt die Gesamtlaufzeit eines Programms sowie die Gesamtzahl der Interrupts, durch die das Programm unterbrochen wurde. Die in einem bestimmten Bereich verbrachte Zeit kann dann nach folgender Formel berechnet werden:

$$\text{Zeit}_{\text{Bereich}} = \text{Zeit}_{\text{gesamt}} * \text{Anzahl}_{\text{Bereich}} / \text{Anzahl}_{\text{gesamt}}$$

Dies ist *nicht* die exakte Zeit, die in einem Bereich zugebracht wurde. Wenn Ihr Programm eine Routine mit einer Frequenz durchläuft, die ein Vielfaches der Frequenz des Zeitgebers ist (zum Beispiel eine Routine, die einen konstanten Ton abgibt), kann es sein, daß die Ausführung einer bestimmten Zeile (oder eines Bereiches) immer genau mit den Interrupts der Zeiterfassung zusammenfällt. Diese Resonanz kann auch dann vorkommen, wenn die Zeile in einem Programmteil steht, in dem das Pro-

programm die meiste Zeit zubringt. So etwas kommt selten vor, ist aber möglich.

Wenn Sie diese Art von Frequenzüberlappung vermuten, sollten Sie die Einstellung des Zeittakts (*speed value*) in *Statistics / Profiling / Options / Clock* ändern und dann das Ergebnis mit einer früheren Analyse vergleichen.

Die Speicherbelegung von Profiler

Der Profiler legt die Bereichsinformationen auf den FAR-Heap. Wenn Sie während des Programmlaufs Bereiche einfügen, wird der FAR-Heap auf Kosten des Benutzerprogramms vergrößert, um für neue Bereichsvariablen und Puffer Platz zu schaffen. Dies ist der Grund, warum Sie das Programm immer mit *Run / Program Reset* zurücksetzen sollten, wenn Sie Bereiche während des Programmlaufs ändern, denn so wird der benutzte Speicher wieder freigegeben. Falls Sie dies unterlassen, sind die Ergebnisse einer Analyse wahrscheinlich falsch. Es kann in Einzelfällen sogar vorkommen, daß sich Ihr System aufhängt, wenn nicht mehr genügend Speicherplatz zur Verfügung steht.

Die Kommandozeilen-Optionen

Das allgemeine Befehlsformat für den Aufruf von Turbo Profiler lautet:

```
TPROF [TProf_Optionen] [Programmname [Programm_Argumente]]
```

wobei *TProf_Optionen* eine Liste von einer oder mehreren Kommandozeilenoptionen für den Profiler darstellt (siehe Tabelle A.1). *Programmname* ist der Name des Programms, das analysiert werden soll.

Programm_Argumente ist eine Liste mit ein oder mehreren Kommandozeilenargumenten des untersuchten Programms.

Sie können TPROF auch ohne Angabe von Programmnamen oder Argumenten starten. In diesem Fall müssen Sie das Programm, das Sie untersuchen wollen, über die Benutzerumgebung von Turbo Profiler laden.

Einige Beispiele für Aufrufe von Turbo Profiler:

<pre>tprof -sc prog1 a b</pre>	ruft den Profiler mit der Option -sc auf und lädt das Programm PROG1 mit den beiden Kommandozeilenargumenten a und b.
<pre>tprof prog2 -x</pre>	startet den Profiler mit den Standardoptionen und lädt das Programm PROG2 mit einem Kommandozeilenargument, -x .

Kommandozeilen-Optionen

Alle Kommandozeilenoptionen von Turbo Profiler werden mit einem Bindestrich eingeleitet. Die Optionen müssen durch mindestens ein Leerzeichen oder einen Tabulator vom Befehl TPROF und anderen Kommandozeilenoptionen getrennt sein.

Um eine Profiler-Option über die Kommandozeile abzuschalten, geben Sie ein Bindestrich hinter der Option an. Zum Beispiel schaltet **-vg-** die

Option *Grafik speichern* ausdrücklich ab. Normalerweise wird eine Option nur dann abgeschaltet, wenn Sie in der Profiler-Konfigurationsdatei `TFCONFIG.TC` aktiviert wurde. Sie können die Konfigurationsdatei mit dem Installationsprogramm `TFINST` ändern. Eine Beschreibung hierzu finden Sie im Anhang B.

Tabelle A.1 faßt alle Kommandozeilenoptionen von Turbo Profiler zusammen. Die Optionen werden auf den nachfolgenden Seiten detailliert erläutert.

Tabelle A.1:
Die Kommandozeilenoptionen von Turbo Profiler

Option	Beschreibung
-b	Lädt Turbo Profiler im Stapelverarbeitungsmodus
-bcAnzahl	Laden im Stapelverarbeitungsmodus mit <i>Anzahl</i> Programmdurchläufen
-cDatei	Einlesen der Konfiguration aus Datei
-do	Umleitung der Profilerausgaben auf einen zweiten angeschlossenen Bildschirm
-dp	Anzeige von Profiler und Programmausgaben auf zwei getrennten Bildschirmseiten
-ds	Getrennte Bildschirmspeicherung für Profiler und Anwendungsprogramm.
-h	Ausgabe von Hilfsinformationen
-?	Ausgabe von Hilfsinformationen
-ji	Protokoll-Speicherung: Protokolldatei nicht verwenden, wenn das Programm neu kompiliert wurde
-jn	Protokoll-Speicherung: Protokolldatei nicht verwenden
-jp	Protokoll-Speicherung: Fragen, ob Protokolldatei verwendet werden soll, wenn das Programm neu kompiliert wurde
-ja	Protokoll-Speicherung: Protokolldatei immer verwenden, auch wenn das Programm neu kompiliert wurde
-p	Mausunterstützung einschalten
-r	Analyse auf einem Remote-System über serielle Verbindung
-rn/r	Analyse auf einem Remote-Netzwerk, wobei <i>/</i> das lokale System bezeichnet und <i>r</i> das Remote-System
-rpn	Ausgabeport <i>n</i> für Remote-Analyse verwenden
-rsn	Übertragungsgeschwindigkeit für das Remote-System festlegen
-sc	Groß-/Kleinschreibung in Symbolnamen vernachlässigen
-sdVerz	Verzeichnisse für Suche nach Quelldateien festlegen
-vg	Zwischenspeichern des gesamten Grafik-Bildschirms
-vn	43/50-Zeilendarstellung ausschalten
-vp	EGA-Palette der Ausgaben zwischenspeichern

Stapelverarbeitungsbetrieb (-b)

Durch diese Option wird Turbo Profiler dazu angewiesen, im Stapelverarbeitungsmodus zu arbeiten. Die Benutzeroberfläche von Turbo Profiler ist im Stapelverarbeitungsmodus nicht aktiv; stattdessen führt Turbo Profiler

das Programm aus und sichert die statistischen Daten in der Datei *Dateiname.TFS*, wobei *Dateiname* der Name der ausführbaren Datei ist.

Mit Hilfe der Option **-b** können Sie eine DOS-Stapelverarbeitungsdatei erzeugen, die ein oder mehrere TPROF-Kommandos enthält. Wenn das Programm, das Sie analysieren, keine Tastatureingaben erfordert, können Sie den Stapelverarbeitungsprozeß laufen lassen und gleichzeitig andere Dinge erledigen.

Hinweis

Bevor Sie Analysen im Stapelverarbeitungsmodus durchführen, müssen Sie eine Bereichsdatei einrichten, um dem Profiler mitzuteilen, welche Bereiche untersucht werden sollen und in welchem Analysemodus er arbeiten soll.

Um eine Bereichsdatei einzurichten, laden Sie Ihr Programm und wählen den gewünschten Analysemodus (*Statistics / Profiling Options*). Als nächstes markieren Sie die Bereiche, die untersucht werden sollen und setzen dabei die erforderlichen Enable- und Disable-Markierungen. Anschließend verlassen Sie Turbo Profiler. Es wird eine .TFA-Datei erzeugt; sie enthält die ausgewählten Bereiche und den Analysemodus.

Wenn diese Bereichsdatei fertig erstellt ist, erzeugen Sie eine DOS-Stapelverarbeitungsdatei mit den gewünschten Turbo-Profiler-Kommandos. Betrachten Sie dazu folgendes Beispiel:

```
TPROF -b MYPROG arg1 arg2
rename MYPROG.TFS MYPROG1.TFS
TPROF -b MYPROG arg1 arg3
rename MYPROG.TFS MYPROG2.TFS
TPROF -b MYPROG arg2 arg2
rename MYPROG.TFS MYPROG3.TFS
TPROF -b MYPROG arg2 arg3
```

Alternativ könnten Sie auch die Anzahl der Programmdurchläufe auf einen Wert größer als 1 setzen, damit alle Statistiken in der gleichen Datei akkumuliert werden.

Beachten Sie, daß Sie die .TFS-Datei nach jedem TPROF-Kommando umbenennen müssen. Wenn Sie das vergessen, werden durch aufeinander folgende Durchläufe des gleichen Programms die bereits erzeugten statistischen Daten überschrieben. Wenn Ihre Stapelverarbeitungsdatei fertig ist, starten Sie diese von der DOS-Kommandozeile aus.

Konfigurationsdatei (-c)

Durch diese Option wird festgelegt, welche Konfigurationsdatei verwendet werden soll. Standardmäßig wird TFCONFIG.TF verwendet. Wenn Sie eine andere Datei laden wollen, müssen Sie direkt (ohne weitere Leerzeichen) hinter der Option **-c** den Namen der gewünschten Konfigurationsdatei angeben.

Darstellungsmodus (-d)

Alle Optionen, die mit **-d** beginnen, beeinflussen die Art und Weise, wie Turbo Profiler die Darstellung behandelt.

- do** leitet die Profiler Ausgaben auf einen zweiten Bildschirm um. Sie können die Ausgaben des Programms auf dem ersten Bildschirm betrachten, während Turbo Profiler den zweiten Bildschirm verwendet.
- dp** Dies ist die Standard-Vorgabe für Farbbildschirme. Damit wird der Profiler auf der Darstellungsseite gespeichert, während die Ausgaben des analysierten Programms auf einer anderen Bildschirmseite dargestellt werden.

Durch den Einsatz von zwei Bildschirmseiten verringert sich der Zeitaufwand zum Umschalten zwischen den beiden Bildschirmdarstellungen. Diese Option kann nur mit Farbgrafik-Adaptern eingesetzt werden, da nur bei diesen mehrere Bildschirmseiten zur Verfügung stehen. Sie können diese Option nicht einsetzen, wenn das untersuchte Programm selbst mehrere Bildschirmseiten verwendet.
- ds** Die Kommandozeilenoption **-ds** ist die Standardeinstellung für Monochrombildschirme. Mit dieser Option werden zwei separate Bildschirmoberflächen aufgebaut: eine für Turbo Profiler und eine andere für das Programm, das zu analysieren ist. Der ganze Bildschirm wird jedesmal aus dem Speicher geladen, wenn zwischen Programm und Profiler umgeschaltet wird. Dies ist zwar die zeitaufwendigste Methode für die Präsentation der beiden Bildschirmoberflächen, dafür ist sie sowohl auf allen Bildschirmen anwendbar als auch mit Programmen, die auf ungewöhnliche Art mit dem Bildschirm arbeiten.

Hilfestellungen (-h und -?)

Mit diesen Optionen erhalten Sie einen Bildschirm mit Hilfsinformationen, in denen die Kommandozeilenoptionen und die Kommandozeilensyntax von Turbo Profiler beschrieben werden.

Protokoll speichern (-j/n)

Mit den Optionen zu dem Sitzungsstatus wird festgelegt, wie Turbo Profiler die Protokolldateien behandeln soll. Weitere Informationen dazu finden Sie auf im Abschnitt „Session Saving“

Mausunterstützung (-p)

Mit dieser Option wird die Mausunterstützung eingeschaltet.

Remote-Analysen (-r)

Alle Optionen, die mit **-r** beginnen, beziehen sich auf den Remote-Betrieb von Turbo Profiler.

- r** Mit der Option **-r** wird die Analyse auf einem System gestartet, das über eine serielle Verbindung angeschlossen ist. Normalerweise werden die Standardeinstellungen für die Schnittstelle (COM1) und Geschwindigkeit (115 KBAud) verwendet, falls diese nicht mit Hilfe von TFINST verändert wurden.
- rnl;r** Analyse über ein lokales Netzwerk; *l* und *r* sind optionale Argumente, die das lokale System bzw. das Remote-System bezeichnen.
- rp*n*** Auswahl der Schnittstelle *n* für die Remote-Verbindung. Bei *n*=1 wird COM1 ausgewählt, bei *n*=2 COM2, bei *n*=3 COM3 und bei *n*=4 COM4
- rs*n*** Mit dieser Option wird die Übertragungsgeschwindigkeit für die Verbindung ausgewählt. Bei *n*=1 wird 9.600 Baud ausgewählt, bei *n*=2 19.200 Baud, bei *n*=3 38.400 Baud und bei *n*=4 115.000 Baud.

Quellcode und Symbole (-s)

Alle Optionen, die mit **-s** beginnen, beziehen sich auf die Behandlung des Quellcodes und der Programmsymbole.

- sc** Ist dieser Schalter gesetzt, wird zwischen Groß- und Kleinschreibung in den Symbolen nicht unterschieden, auch wenn das Programm unter Berücksichtigung dieser Unterschiede gelinkt wurde.

Ohne die Option **-sc** vernachlässigt Turbo Profiler die Groß- und Kleinschreibung nur dann, wenn Sie Ihr Programm mit der Option »Groß- und Kleinschreibung ignorieren« kompiliert haben.
- sd*Verz*** Auswahl von einem oder mehreren Verzeichnissen, die nach Quelldateien durchsucht werden sollen. Die Syntax lautet:

-sdVerzeichnis

Verzeichnis kann ein relativer oder absoluter Pfad sein und eine Laufwerksbezeichnung enthalten. Um mehrere Verzeichnisse festzulegen, verwenden Sie die Option **-sd** mehrere Male, je einmal pro Verzeichnis, oder Sie geben hinter einer einzigen Anweisung der Art **-sd** mehrere Verzeichnisse an, die voneinander durch Semikola getrennt sind:

-sdVerz1;Verz2;Verz3

Turbo Profiler durchsucht die Verzeichnisse in der angegebenen Reihenfolge.

Wenn in der Konfigurationsdatei eine Verzeichnisliste enthalten ist, werden die mit **-sd** angegebenen Verzeichnisse angehängt.

Bildschirm-Optionen (-v)

Alle Optionen, die mit **-v** beginnen, beziehen sich auf die Ansteuerung der Bildschirm-Adapter.

- vg** speichert den gesamten Grafikbildschirm eines Programms. Dazu ist zusätzlicher Speicher nötig. Sie können dafür aber auch Programme untersuchen, die in speziellen Grafik-Modi arbeiten. Versuchen Sie diese Option, wenn die Bildschirmausgabe unter Turbo Profiler nicht wie gewünscht funktioniert.
- vn** schaltet den 43/50-Zeilenmodus ab. Durch die Angabe dieser Option können Sie etwas Speicher einsparen. Verwenden Sie **-vn**, wenn Sie mit einem EGA- oder VGA-Bildschirm arbeiten und von vornherein wissen, daß Sie unter Turbo Profiler nicht im 43- oder 50-Zeilenmodus arbeiten wollen.
- vp** schaltet das Speichern der EGA/VGA-Palette der Programmausgaben ab. Verwenden Sie diese Option für Programme, die auf spezielle EGA-/VGA-Modi umschalten.

Die Installation von Turbo Profiler

Turbo Profiler ist sofort nach der Kopie der Dateien von den Originaldisketten lauffähig. Mit Hilfe des Installationsprogramms TFINST können Sie jedoch viele der Standardeinstellungen ändern. Einige der Optionen lassen sich auch beim Aufruf von Turbo Profiler mit Hilfe von Kommandozeilenoptionen ändern. Sollten Sie feststellen, daß Sie immer wieder dieselben Optionen benötigen, empfiehlt es sich, diese Optionen mit Hilfe des Installationsprogramms festzuschreiben.

Im Installationsprogramm sind folgende Einstellungen möglich:

- Farben und Hintergrundmuster für Bildschirm und Fenster;
- Darstellungsparameter: Bildschirm-Modus, Anzahl der Zeilen, Spaltenbreite für Tabulatoren, schnelle Bildschirmanpassung, 43-/50-Zeilenmodus, Zwischenspeichern des kompletten Grafikbildschirms, Darstellung der Ausgaben des Benutzerprogramms;
- Aufruf des Editors; Verzeichnisse für Quelldateien sowie für die Hilfe- und Konfigurationsdateien von Turbo Profiler;
- Parameter für Eingaben und Eingabeaufforderungen: Länge der Eingabeauffzeichnungsliste, Alarmton bei Fehlern, Maus, Aufnahme von Tastatureingaben und Tastenkürzel für Control-Tasten;
- Quellcode-Analyse: Optionen für Programmiersprachen und Unterscheidung zwischen Groß- und Kleinschreibung;
- NMI-Intercept und Remote-Analysen;
- Darstellungsmodus.

Der Aufruf von TFINST

Für den Start des Installationsprogramms geben Sie in der DOS-Ebene **TFINST** ein. Daraufhin erscheint das Hauptmenü von TFINST. Zur Auswahl eines Menüeintrags drücken Sie einen der hervorgehoben dargestellten Buchstaben der Optionen oder verwenden Sie die Pfeiltasten [*Nach oben*] oder [*Nach unten*], um die Option auszuwählen, und drücken Sie dann [*Eingabe*]. Drücken Sie zum Beispiel [*D*], um die Bildschirmeinstellungen (*Display*) zu ändern. Die weiteren Menüs des Installationsprogramms werden ebenso bedient. Für die Rückkehr zum übergeordneten Menü drücken Sie [*Esc*], und wiederholen dies gegebenenfalls, um zum Hauptmenü zurückzukehren.

Zum Verlassen von TFINST wählen Sie im Hauptmenü *File / Quit* (oder [*A*]/[*X*]).

Einstellung der Bildschirmfarben

Wählen Sie im Hauptmenü *Colors*. Sie erhalten zwei Optionen zur Auswahl: *Customize* und *Default Color Set*.

Auswahl der Bildschirmfarben

Fenster

Nach Auswahl von *Customize* erscheint ein drittes Menü zur Anpassung der Fenster, Dialogfenster, Menüs und Bildschirme.

Zur Anpassung der Fenster wählen Sie den Befehl *Windows*. Dadurch wird ein viertes Menü geöffnet, in dem Sie das Fenster auswählen, das angepaßt werden soll: *Text*, *Statistics* oder *Disassembly* (das CPU-Fenster). Nach Auswahl einer dieser Optionen erscheint ein weiteres Menü, das die möglichen Fensterelemente enthält. Ferner erscheint ein Paar von zwei Beispielfenstern (von denen nur eins aktiv ist und das andere nur als Beispiel dient, also inaktiv ist), mit denen Sie die verschiedenen Farbkombinationen ausprobieren können. Der Bildschirm sieht so aus:

Abbildung B.1:
Anpassung der Fensterfarben



Wenn Sie ausgewählt haben, was Sie verändern wollen, erscheint eine Palette über dem Menü. Mit Hilfe der Pfeiltasten können Sie die gewünschte Kombination auswählen. Wenn Sie die Farbzusammenstellung ändern, wird sofort die aktuelle Kombination angezeigt. Haben Sie die gewünschte Farbe gefunden, drücken Sie [*Eingabe*], um die Farbe zu übernehmen.

Hinweis

Turbo Profiler verwaltet drei Farbtabelle: eine für Farbe, eine für Schwarzweiß und eine für Monochrom. Sie können zu einem bestimmten Zeitpunkt nur eine der drei Farben ändern, je nachdem, welcher Videomodus eingestellt ist und welche Hardware vorliegt. Wenn Sie mit einem Farbmonitor arbeiten und die Schwarzweiß-Tabelle anpassen wollen, setzen Sie den Videomodus auf Schwarzweiß, indem Sie den Befehl **MODE BW80** nach der DOS-Eingabeaufforderung eingeben und danach TFINST starten.

Dialogfenster und Menüs

Wenn Sie *Dialogs* oder *Menus* im Menü *Customize* auswählen, erscheint ein Dialogfenster, in dem weitere Menüs oder Menüpunkte aufgelistet sind, sowie ein Beispiel für ein Dialogfenster oder ein Menü, mit dem Sie experimentieren können.

Wie im Menü *Windows* wird bei der Auswahl eines Menüpunkts im aktuellen Menü ein Fenster mit einer Farbpalette geöffnet, aus der Sie die Farbe für diesen Eintrag auswählen können.

Bildschirmelemente

Die Auswahl von *Screen* im Menü *Customize* öffnet ein Menü, in dem Sie auf ein weiteres Menü mit Hintergrundmustern und Paletten für Bildschirmelemente zugreifen können, sowie ein Beispielhintergrund zum Ausprobieren.

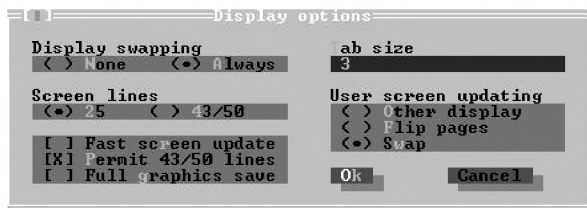
Die vorgegebene Farbkombination

Wenn Sie im Menü *Colors* die Option *Default Color Set* auswählen, wird ein aktives Textfenster und ein nicht aktives Fenster mit den Standardfarben auf dem Bildschirm angezeigt. Mit einem Dialogfenster können Sie sich die Farbeinstellungen für das Text-, Statistik- oder Disassembler-Fenster anzeigen lassen.

Die Bildschirmparameter von Turbo Profiler

Nach Auswahl der Option *Display* im Hauptmenü erhalten Sie das Dialogfenster *Display Options*.

Abbildung B.2:
Das Dialogfenster
Display Options



Hinweis

Einige dieser Darstellungsoptionen können Sie auch beim Aufruf von Turbo Profiler über die DOS-Kommandozeile einstellen. Es gibt jedoch auch einige Einstellungen, die nur in TFINST geändert werden können. In Tabelle A.1 finden Sie eine Liste der Kommandozeilenoptionen von Turbo Profiler und deren Entsprechungen in TFINST.

**Bildschirmum-
schaltung (Dis-
play Swapping)**

Mit Hilfe der Auswahlfelder von *Display Swapping* wird festgelegt, wie Turbo Profiler zwischen dem eigenen Bildschirm und dem Bildschirm des untersuchten Programms umschaltet. Sie können zwischen den folgenden Einstellungen wählen:

- None* Zwischen den beiden Bildschirmen wird niemals umgeschaltet. Verwenden Sie diese Option, wenn Sie ein Programm untersuchen, das keine Ausgaben auf den Bildschirm schreibt.
- Always* Es wird jedesmal auf den Ausgabebildschirm umgeschaltet, wenn das Anwenderprogramm läuft. Verwenden Sie diese Option, wenn Ihr Programm Bildschirmausgaben macht. Dies ist auch die Standardeinstellung.

Screen Lines

Mit Hilfe dieser Auswahlfelder können Sie zwischen einer Darstellung von 25 Zeilen und 43 bzw. 50 Zeilen umschalten.

Hinweis

Nur auf einem EGA- oder VGA-Bildschirm können mehr als 25 Zeilen dargestellt werden.

**Fast Screen
Update**

Mit dem Markierungsfeld *Fast Screen Update* können Sie festlegen, ob die Bildschirmausgaben schnell oder verzögert erfolgen sollen. Schalten Sie diese Einstellung um, wenn es auf Ihrem Bildschirm durch „Schnee“ zu Bildschirmstörungen kommen sollte. Diese Option brauchen Sie nur dann auszuschalten, wenn Sie der „Schnee“ stört. (Manche Leute bevorzugen die „gestörte“ Anzeige, da diese wesentlich schneller ist.)

**Permit 43/50
Lines**

Wenn dieses Markierungsfeld eingeschaltet ist, kann man auf eine Darstellung mit 43/50-Zeilen umschalten. Wenn Sie diese Einstellung ausschalten, sparen Sie etwa 8 KB an Speicherplatz ein, da die aufwendigeren Bildschirmdarstellungen einen größeren Pufferbereich beanspruchen. Dies ist vor allem dann sehr praktisch, wenn Sie ein sehr großes Programm untersuchen, das für die Ausführung soviel Speicher wie möglich benötigt. Wenn diese Option ausgeschaltet ist, können Sie nicht in den 43/50-Zeilenmodus umschalten, auch wenn Ihr System dazu in der Lage ist.

Full Graphics Saving

Schaltet man dieses Markierungsfeld ein, wird jedesmal der gesamte Grafikpuffer gespeichert, wenn zwischen Turbo Profiler und dem Ausgabebildschirm umgeschaltet wird. Wenn Sie dieses Markierungsfeld ausschalten, können Sie ungefähr 12 KB Speicherplatz einsparen. Dies ist besonders hilfreich, wenn Sie ein großes Programm untersuchen, das für die Ausführung soviel Speicher wie möglich benötigt. Wenn man diese Option ausschaltet, liegt der einzige Nachteil im allgemeinen nur darin, daß ein kleiner Bereich des Ausgabebildschirms zerstört wird, was die Analyse jedoch nicht stört.

Tab size

In diesem Eingabefeld können Sie die Anzahl von Spalten festlegen, die zwischen den Tabulator-Positionen in einem Text oder in einer Quelldatei liegen. Sie werden nach der Anzahl an Spalten gefragt (zwischen 1 und 32); die Vorgabe beträgt 8.

User Screen Updating

In den Auswahlfeldern von *User Screen Updating* wird festgelegt, wie der Ausgabebildschirm des Anwendungsprogramms aufgefrischt wird, wenn zwischen Ausgabebildschirm und Programm umgeschaltet wird. Es sind drei Einstellungen möglich:

Other Display Turbo Profiler wird auf einem zweiten Bildschirm in Ihrem System dargestellt. Wenn Sie sowohl einen Farb- als auch einen Monochrome-Adapter haben, können Sie Ihr Programm auf dem einen Bildschirm und die Analysen von Turbo Profiler auf dem anderen gleichzeitig darstellen.

File Pages Verlegt die Bildschirmoberfläche von Turbo Profiler auf eine separate Bildschirmseite. Mit dieser Option können Sie also nur arbeiten, wenn Ihr Bildschirmadapter über verschiedene logische Bildschirmseiten verfügt, wie dies zum Beispiel bei CGA-, EGA- oder VGA-Bildschirmadaptern möglich ist. Bei einem Monochrombildschirm kann diese Option nicht verwendet werden. Diese Option arbeitet sehr schnell, stört allerdings manchmal die Ausführung von Programmen, die mehrere logische Bildschirmseiten verwenden. Glücklicherweise sind solche Programme aber recht selten.

Swap Benützt einen einzigen Bildschirmadapter und eine Bildschirmoberfläche. Die Inhalte des Benutzerbildschirms und des Turbo-Profiler-Bildschirms werden immer zwischen Hauptspeicher und Bildschirmadapter hin- und hergeladen. Dies ist die langsamste Methode des Bildschirmwechsels, aber auch die sicherste und diejenige, die das Benutzerprogramm am geringsten beeinflusst. Wählen Sie diese

Option, wenn Sie ein Programm mit mehreren logischen Bildschirmseiten analysieren. Die Option *Swap* dient ebenso dazu, beim Aufruf von speicherresidenten Programmen oder Programmen über die DOS-Shell-Funktion den aktuellen Turbo-Profiler-Bildschirm wiederherzustellen.

Die Optionen von Turbo Profiler

Der Befehl *Options* des Hauptmenüs öffnet ein Menü mit Optionen, die zu Dialogfenstern führen.

Das Dialogfenster Directories

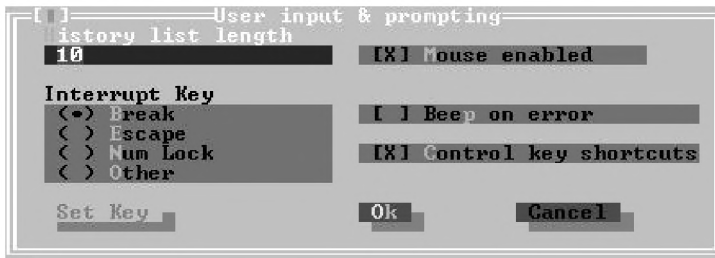
Dieses Dialogfenster enthält drei Eingabefelder für folgende Angaben:

- | | |
|---------------------|---|
| Editor program name | In diesem Eingabefeld wird der DOS-Befehl festgelegt, mit dem Ihr Editor aufgerufen wird. Damit können Sie aus Turbo Profiler heraus Ihren gewohnten Editor aufrufen, falls Sie während der Untersuchung eine Datei ändern wollen. Turbo Profiler hängt an diesen Befehl durch ein Leerzeichen abgetrennt den Namen der Datei an, die Sie editieren wollen. |
| Source directories | In diesem Feld können Sie eine Liste von Verzeichnissen angeben, in denen Turbo Profiler nach Quelldateien sucht. |
| Turbo directory | Legt das Verzeichnis fest, in dem Turbo Profiler nach den Dateien mit den Hilfestellungen und der Konfiguration sucht. |

Das Dialogfenster User Input & Prompting

In diesem Dialogfenster legen Sie fest, wie Informationen in Turbo Profiler eingegeben werden können und wie Turbo Profiler weitere Informationen anfordert.

Abbildung B.3:
Das Dialogfenster
User Input and
Prompting



History List Length

In diesem Eingabefeld geben Sie an, wieviele der früheren Eingaben in der Eingabeaufzeichnungsliste eines Eingabefelds gespeichert werden sollen.

Interrupt Key

In diesem Feld können Sie die Voreinstellung für die Unterbrechungstaste von Turbo Profiler von der Voreinstellung *Break* in jede andere Taste oder Tastenkombination ändern. Wenn Sie den Aktionsschalter *Other* betätigen, erscheint eine Eingabeaufforderung; anschließend drücken Sie die Taste oder die Tastenkombination, die die Funktion der Unterbrechungstaste übernehmen soll.

Mouse Enabled

Mit diesem Markierungsfeld bestimmen Sie, ob Turbo Profiler standardmäßig eine Maus unterstützen soll.

Beep on Error

Standardmäßig reagiert Turbo Profiler mit einem Alarmton, wenn Sie eine unzulässige Taste drücken oder eine Fehlermeldung verursachen. Mit dem Markierungsfeld *Beep on Error* können Sie diese Vorgabe ändern.

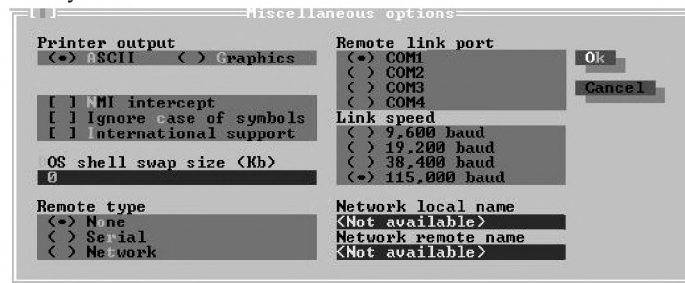
Control Key Shortcuts

Die Tastenkürzel für Control-Tastenbefehle lassen sich mit diesem Markierungsfeld ein-/ausschalten. Wenn die Control-Tastenkürzel aktiviert sind, können Sie alle lokalen Menübefehle direkt über die Taste [Ctrl] und dem Anfangsbuchstaben des Menüpunkts aufrufen. Dadurch ist es nicht mehr möglich, diese Control-Tasten im Stil der Wordstar-Cursor-Befehle zum Bewegen im Text zu verwenden.

Das Dialogfenster Miscellaneous Options

Das Dialogfenster *Miscellaneous Options* enthält Optionen für die Einstellung von Interrupts, EMS-Speicher, DOS-Auslagerungen und Remote-Analysen.

Abbildung B.4:
Das Dialogfenster
*Miscellaneous
Options*



Printer Output

Mit dieser Option schalten Sie zwischen dem erweiterten IBM-ASCII-Zeichensatz und dem Standard-ASCII-Zeichensatz für Druckerausgaben um.

NMI-Intercept

Dieser nicht maskierbare Interrupt (d.h. dieser Interrupt kann per Software nicht ausgeschaltet werden) ist ein Hardware-Interrupt, den der Prozessor ohne Verzögerung bearbeiten muß. Er wird typischerweise dazu benutzt, Ihr System anzuhalten, wenn ein Paritätsfehler im Speicher aufgetreten ist: in einem solchen Fall wird auf Ihrem System eine Meldung wie Memory Parity Error ausgegeben – danach hängt sich Ihr Rechner auf.

Dieser Interrupt wird allerdings auch dazu verwendet, einen Programmabbruch durch ein Debugger-Board zu erzeugen, wenn eine Abbruchtaste gedrückt wurde. Da die NMI-Voreinstellung in Turbo Profiler auf *Off* gesetzt ist, werden Sie diese Einstellung möglicherweise ändern wollen, wenn Sie ein Debugger-Board benutzen.

Wenn es sich bei Ihrem Computer nicht um einen Tandy 1000A, IBM PC Convertible, ACER 1100 oder NEC MultiSpeed handelt, können Sie mit TFINST das Markierungsfeld *NMI Intercept* auf *On* schalten. Einige Computersysteme verwenden den NMI in einer Art und Weise, die mit Turbo Profiler nicht verträglich ist, so daß Sie die Verarbeitung dieses Interrupts in Turbo Profiler ausschalten müssen, damit das andere Programm läuft.

Ignore Case of Symbol

Wenn dieses Markierungsfeld eingeschaltet ist, unterscheidet Turbo Profiler nicht zwischen Groß- und Kleinschreibung.

International support

Mit diesem Markierungsfeld wird Turbo Profiler dazu angewiesen, Ausdrücke in Fenstern, die Auflistungen enthalten, entsprechend des Alphabets zu sortieren, das aufgrund des Country-Eintrags in Ihrer CONFIG.SYS-Datei aktiv ist (unter DOS) oder entsprechend der Länderkennung, die über das Windows Control Panel aktiv ist. Weitere Informationen über das Einstellen der Länderkennung finden Sie in Ihrem DOS- beziehungsweise Windows-Handbuch.

Falls Sie in diesem Feld keine Einstellung vornehmen, wird die Sortierreihenfolge unter DOS durch die ASCII-Werte der Ausdrücke, unter Windows durch deren ANSI-Werte festgelegt.

DOS Shell Swap Size (KByte)

In diesem Eingabefeld können Sie die Anzahl an KBytes angeben, die für den Befehl *File / DOS Shell* reserviert werden sollen, damit dieser Befehl auch nach dem Laden größerer Programme noch verwendet werden kann.

Diese Option steht bei TPROFW nicht zur Verfügung.

Remote Type

Dieses Markierungsfeld ist für die Art der Verbindung bei Remote-Analysen zuständig. Die Voreinstellung lautet hier *None*, was bedeutet, daß die Analyse lokal durchgeführt und keine Remote-Verbindung aufgebaut wird. Bei der Einstellung *Serial* findet eine Remote-Analyse mit Hilfe einer seriellen Verbindung statt; die Datenübertragungsschnittstelle und die Übertragungsgeschwindigkeit werden durch die Optionen in Remote Link Port und Link Speed festgelegt. Das Markierungsfeld *Network* richtet eine Remote-Analyse über ein LAN ein.

Achtung

Es ist normalerweise nicht angebracht, eine Konfigurationsdatei für eine Remote-Analyse zu sichern, da Turbo Profiler in diesem Fall bei jedem weiteren Einsatz diese Remote-Verbindung aufbaut.

Remote Link Port

In den Auswahlfeldern für die Schnittstelle der Remote-Verbindung können Sie zwischen COM1 und COM2 wählen.

Link Speed

Mit den Auswahlfeldern *Link Speed* können Sie zwischen vier verfügbaren Übertragungsgeschwindigkeiten wählen: 9600 Baud, 19200 Baud, 38400 Baud und 115000 Baud.

Network local name

In diesem Textfeld können Sie einen Namen für das Lokale System eingeben, wenn Sie eine Remote-Analyse über ein LAN durchführen. Aufgrund der Voreinstellung lautet der Name für das lokale System LOCAL. Sie sollten diesen Namen ändern, wenn mehr als eine Person das Netzwerk für Remote-Analysen verwendet.

Network remote name

Wenn Sie Remote-Analysen über ein LAN durchführen, definieren Sie in diesem Textfeld den Namen des Remote-Systems. Per Voreinstellung lautet dieser Name REMOTE. Sie sollten diesen Namen ändern, wenn mehr als eine Person das Netzwerk für Remote-Analysen verwendet.

Einstellung des Bildschirmmodus

Nach Auswahl von *Mode for Display* im Hauptmenü wird ein Menü geöffnet, in dem Sie den Bildschirmmodus Ihres Systems auswählen können.

Default

Turbo Profiler ermittelt den Typ Ihres Grafik-Adapters selbständig und wählt den dafür geeigneten Bildschirmmodus aus.

Color

Wenn Sie einen EGA-, VGA-, CGA-, MCGA- oder 8514-Adapter haben und diese Einstellung auswählen, erfolgt die Darstellung in Farbe.

Black & White

Wenn Sie einen EGA-, VGA-, CGA-, MCGA- oder 8514-Adapter haben und diese Einstellung auswählen, erfolgt die Darstellung in Schwarzweiß.

Wenn Sie mit den Einstellungen fertig sind

Monochrome

Wählen Sie diese Option aus, wenn Sie einen Farb-Monitor zusammen mit einer Hercules- oder Monochrome-Karte verwenden, die nur über Textmodus verfügt.

LCD

Durch Auswahl dieser Option erhalten Sie für LCD-Bildschirme eine besser lesbare Darstellung als mit *Black & White*.

Wenn Sie mit den Einstellungen fertig sind

Nachdem Sie Turbo Profiler konfiguriert und angepaßt haben, werden Sie den Wunsch haben, diese Einstellungen auf Diskette zu speichern. Sie können zu diesem Zweck das ausführbare Programm des Turbo Profiler (TPROF.EXE) direkt modifizieren oder eine Konfigurationsdatei erzeugen, die zusammen mit Turbo Profiler geladen wird.

Speichern der Änderungen

Wenn Sie alle Optionen von Turbo Profiler wie gewünscht eingestellt haben, wählen Sie den Eintrag *Save* im Hauptmenü, um festzulegen, wie sie gespeichert werden sollen.

Save Configuration File

Wenn Sie die Option *Save Configuration File* auswählen, wird ein Dialogfenster geöffnet, in dem der Name der Konfigurationsdatei TFCON-FIG.TF als Vorgabe erscheint. Sie können diesen Namen mit *[Eingabe]* übernehmen oder einen anderen Namen für die Konfigurationsdatei eingeben. Wenn Sie einen anderen Dateinamen eingeben, können Sie diese Konfigurationsdatei mit Hilfe der Kommandozeilenoption **-c** beim Aufruf von Turbo Profiler auswählen. Zum Beispiel:

```
tprof -cspezial programm
```

Die Datei kann aber auch mit dem Befehl *Options / Restore Configuration* während der Arbeit mit Turbo Profiler eingelesen werden.

Modify TPROF.EXE

Wenn Sie *Modify TPROF.EXE* auswählen, werden alle Konfigurationsänderungen direkt in der Programmdatei von Turbo Profiler, TPROF.EXE, gespeichert. Beim nächsten Aufruf von Turbo Profiler werden diese Einstellungen standardmäßig verwendet.

Wenn Sie später wieder auf die Standardeinstellung zurückgreifen wollen, mit der Turbo Profiler ausgeliefert wird, kopieren Sie TPROF.EXE von den Original-Disketten auf Ihr Arbeitssystem und überschreiben so die geänderte Version.

TFINST verlassen

Sie können das Programm TFINST jederzeit verlassen, indem Sie im Hauptmenü den Befehl *Quit* auswählen.

Kommandozeilenoptionen und die entsprechenden TFINST-Optionen

Einige der beschriebenen Optionen können durch Kommandozeilenoptionen aufgehoben werden. Tabelle B.1 zeigt die Entsprechungen zwischen den Kommandozeilenoptionen von TPROF und den Optionen von TFINST.

Tabelle B.1:
Die Entsprechungen
der Kommandozeilen-
optionen

Option	Beschreibung
	Display / Display Options
-do	(.) Other Display
-dp	(.) Flip Pages
-ds	(.) Swap
	Options / Input, Prompting / User Input, Prompting
-p	[X] Mouse enabled
-p-	[] Mouse enabled
	Options / Miscellaneous / Miscellaneous Options
-r-	(.) None
-r	(.) Serial
-rn	(.) Network
	Options / Miscellaneous / Miscellaneous Options
-rp1	(.) COM1
-rp2	(.) COM2
	Options / Miscellaneous / Miscellaneous Options
-rs1	(.) 9600 Baud
-rs2	(.) 18200 Baud
-rs3	(.) 38400 Baud
-rs4	(.) 115000 Baud
	Options / Miscellaneous / Miscellaneous Options
-sc	[X] Ignore Case of Symbol
-sc-	[] Ignore Case of Symbol
	Options / Directories / Directories
-sd	Quellcode-Verzeichnisse
	Options / Miscellaneous / Miscellaneous Options
-vn	[] 43/50 Zeilen zulassen
-vn-	[X] 43/50 Zeilen zulassen

TFINST.EXE verwendet folgende Syntax:

```
TFINST [Optionen] [EXE-Datei]
```

Die Ausdrücke in den Klammern sind optional. Die folgende Tabelle führt die Optionen auf:

Tabelle B.2:
Optionen für
TFINST.EXE

Option	Beschreibung
-c <i>Datei</i>	Konfigurationsdatei <i>Datei</i> verwenden
-h, -?	Hilfeinformationen anzeigen
-p	Mausunterstützung aktivieren
-w	TPROFW.EXE konfigurieren

Remote-Analysen

Die Bezeichnung Remote-Analyse (remote = engl. entfernt) erklärt sich selbst: Sie können Turbo Profiler auf einem Rechner laufen lassen und das zu analysierende Programm auf einem anderen. Die Systeme können über serielle Schnittstellen verbunden sein, aber auch über ein NETBIOS-kompatibles lokales Netzwerk (LAN - *Local Area Network*).

Eine Remote-Analyse ist in mehreren Situationen nützlich:

- Wenn das zu untersuchende Programm viel Speicherplatz benötigt und es nicht möglich ist, das Programm und gleichzeitig Turbo Profiler auf ein- und demselben System zu betreiben. Wenn das der Fall ist, erscheint die Meldung *Not enough memory* (engl. zuwenig Speicherplatz).
- Wenn das Programm unter Turbo Profiler zwar noch geladen werden kann, der Speicherplatz jedoch für eine ordnungsgemäße Ausführung des Programms nicht mehr reicht. In diesem Fall werden während der Analyse Fehlermeldungen erscheinen, die sich auf die Speicherzuordnung beziehen.
- Wenn Sie ein Windows-Programm untersuchen.

Wenn Sie eine Windows-Anwendung untersuchen, können Sie entweder Turbo Profiler für Windows (TPROFW) und die Anwendung auf einem Computer laufen lassen oder WREMOTE und die Anwendung auf einem und Turbo Profiler auf einem anderen Rechner.

Obwohl es viele Gründe gibt, warum Sie ein Programm unter Verwendung von zwei Systemen analysieren möchten, sind die Vorteile noch größer, wenn Sie Anwendungen für Windows entwickeln:

- Wenn Sie nur einen Monitor verwenden und Turbo Profiler und die Anwendung auf einem Rechner laufen lassen, müssen Sie zwischen den Bildschirmen von TPROFW und den Bildschirmen der Anwendung hin- und herschalten.

Mit WREMOTE können Sie den Bildschirm der Anwendung und den von Turbo Profiler gleichzeitig beobachten. (Mit zwei an einem Computer angeschlossenen Bildschirmen erzielen Sie das gleiche Ergebnis.)

- TFREMOTE und WREMOTE brauchen viel weniger Speicher als Turbo Profiler, so daß das Programm sich eher so verhält, wie es normal (ohne Turbo Profiler im Hintergrund) läuft.

Bevor Sie sich aufgrund von Speicherplatzproblemen zur Remote-Analyse entschließen, sollten Sie sicherstellen, daß Turbo Profiler den ganzen vorhandenen EMS-Speicher verwenden kann. Mit dem Konfigurationsprogramm TFINST Können Sie EMS-Speicher für die Symboltabelle des Programms zur Verfügung stellen.

Voraussetzungen bei Hard-und Software

Für die Remote-Analyse verwenden Sie entweder eine serielle oder eine LAN-Verbindung. Diese beiden Vorgehensweisen stellen verschiedene Anforderungen an Hard- und Software; folgende Voraussetzungen sind allerdings in beiden Fällen nötig:

- ein Entwicklungssystem mit genug Speicherplatz, um Turbo Profiler laden zu können (betrifft das lokale System) und
- ein zweiter PC (das Remote-System) mit ausreichenden Reserven an Arbeits- und Plattenspeicher, um TFREMOTE und das zu untersuchende DOS-Programm zu laden, bzw. WREMOTE, Windows und das Windows-Programm, das Sie untersuchen wollen.

Wenn Sie eine Windows-Anwendung untersuchen, muß das Remote-System dazu in der Lage sein, im Protected-Mode zu laufen (d. h. es muß über eine CPU vom Typ 80286 oder ein Nachfolgemodell verfügen). Der Betrag an notwendigem Speicherplatz hängt von der Windows-Betriebsart ab; es muß jedoch mindestens 1 MB zur Verfügung stehen.

Bei einer seriellen Verbindung benötigen Sie ein NULL-Modem-Kabel, um die beiden Systeme miteinander zu verbinden. Prüfen Sie zunächst, ob das Kabel, mit dem Sie die beiden Systeme verbinden wollen, korrekt angeschlossen ist. Ein einfaches Verlängerungskabel reicht hier nicht. Beim Kabel müssen zumindest die Datenleitungen zum Senden (TD, Transmit) und Empfangen (RD, Receive) gekreuzt sein. Bei einem 25-poligen Kabel sind das die Leitungen 2 und 3.

Bei einer LAN-Verbindung benötigen Sie außerdem Software, die zu Netware von Novell kompatibel ist (IPX und NETBIOS Version 3.0 oder später).

Hinweis

NETBIOS muß sowohl auf dem lokalen als auch auf dem Remote-System geladen sein, bevor Turbo Profiler oder WREMOTE geladen werden kann. Das gilt für die normale Analyse und auch für die Analyse unter Windows.

Remote-Analysen von DOS-Anwendungen

Bei der Remote-Analyse einer DOS-Anwendung laufen TFREMOTE und die Anwendung auf dem einen Rechner, während Turbo Profiler auf dem zweiten Rechner läuft. In dieser Beschreibung wird der Rechner, auf dem TFREMOTE und die Anwendung laufen, als *Remote-Rechner* bezeichnet und der andere Rechner, auf dem Turbo Profiler läuft, als *lokaler Rechner*.

**Das Remote-Sy-
stem einrichten**

Kopieren Sie das Treiberprogramm TFREMOTE.EXE auf das *Remote-System*. Dorthin müssen Sie auch alle Dateien kopieren, die das Programm, das untersucht werden soll, benötigt. Dazu gehören alle Daten-, Konfigurations-, Hilfedateien usw. Wenn Sie wollen, können Sie auch das Programm, das Sie untersuchen wollen, auf das *Remote-System* übertragen. Das ist aber nicht unbedingt nötig, da Turbo Profiler diese Datei beim Laden gegebenenfalls über die serielle Schnittstelle überträgt.

Sie können die Dateien mit Hilfe von Disketten übertragen oder dazu das Hilfeprogramm TDRF verwenden.

**TFREMOTE
einrichten**

Sie müssen das Treiberprogramm TFREMOTE beim Start mit Hilfe von Kommandozeilenoptionen so einrichten, daß es über die Schnittstelle kommunizieren kann. Die Optionen beginnen mit einem Bindestrich (-) oder mit einem Schrägstrich (/).

Tabelle C.1:
Die Kommandozei-
lenoptionen von
TFREMOTE

Option	Beschreibung
-? oder -h	Hilfeinformationen ausgeben
-rn <i>Remote-Name</i>	Remote-Analyse mit Hilfe eines LANs
-rp1	Schnittstelle 1, (COM1); Standardeinstellung
-rp2	Schnittstelle 2, (COM2)
-rp3	Schnittstelle 3, (COM3)
-rp4	Schnittstelle 4, (COM4)
-rs1	Langsamste Übertragungsgeschwindigkeit, 9600 Baud
-rs2	Langsame Übertragungsgeschwindigkeit, 19200 Baud
-rs3	Mittlere Übertragungsgeschwindigkeit, 38400 Baud
-rs4	Hohe Übertragungsgeschwindigkeit, 115000 Baud (Standardeinstellung)
-w	Optionen in der Programmdatei speichern

Hinweis

Um eine Liste mit den für TFREMOTE verfügbaren Kommandozeilenoptionen zu erhalten, geben Sie den Programmnamen gefolgt von *-h* ein:

```
TFREMOTE -h
```

TFREMOTE anpassen

Wenn Sie TFREMOTE ohne Kommandozeilenoptionen aufrufen, werden die in der Programmdatei voreingestellte Schnittstelle und Baudrate (COM1 und 115000 Baud) verwendet, falls diese nicht mit der Option **-w** verändert wurden.

Die Kommandozeilenoptionen von TFREMOTE kann man für alle weiteren Aufrufe in der Programmdatei von TFREMOTE speichern. Erledigen Sie dies mit der Kommandozeilenoption **-w**, die Sie zusammen mit den anderen zu speichernden Optionen angeben sollten. Sie werden anschließend nach dem Namen der Programmdatei gefragt; wenn Sie den Namen einer ausführbaren Datei eingeben, die nicht existiert, erzeugt TFREMOTE eine Datei dieses Namens. Wenn Sie [*Eingabe*] drücken, wird das gerade laufende Programm (normalerweise handelt es sich dabei um TFREMOTE.EXE) überschrieben.

Wenn Sie mit MS DOS 3.0 oder einer der nachfolgenden Versionen arbeiten, erhalten Sie automatisch in der Eingabeaufforderung den Verzeichnispfad und den Namen der Datei, mit der TFREMOTE aufgerufen wurde. Sie können diese Angabe sofort mit der Eingabetaste [*Eingabe*] bestätigen oder einen anderen Namen für die Programmdatei eingeben (Wenn Sie unter DOS 2.xx arbeiten, müssen Sie den vollständigen Verzeichnis- und Dateinamen des Programms angeben).

Geben Sie beispielsweise auf dem Remote-System nach der DOS-Eingabeaufforderung folgende Zeile ein:

```
TFREMOTE -w -rs3 rp2
```

Nach der entsprechenden Eingabeaufforderung geben Sie den Namen für das modifizierte Programm ein, also etwa TFREMOT2.EXE. Dann erzeugt TFREMOTE einen neuen Remote-Treiber mit dem Namen TFREMOT2.EXE; per Voreinstellung gelten dann für diesen Treiber die Übertragungsrate 38400 Baud (**-rs3w**) und die Schnittstelle COM2 (**-rp2**).

Der Remote- Treiber für DOS

Wenn Sie mit der Remote-Analyse beginnen, müssen Sie zuerst den Treiber auf dem Remote-System starten und anschließend Turbo Profiler auf dem lokalen System laden.

Dabei ist folgendes zu beachten: Aktuelles Verzeichnis auf dem Remote-System muß das Verzeichnis sein, das die zu untersuchenden Dateien enthält. Dies ist wichtig, denn TFREMOTE legt das zu untersuchende Programm in das Verzeichnis, das zu dem Zeitpunkt aktuell war, als TFREMOTE gestartet wurde.

TFREMOTE erscheint mit einer Copyright-Notiz und zeigt an, daß es auf den Start von Turbo Profiler an dem anderen Ende der Leitung wartet. Wenn Sie anhalten und zu MS-DOS zurückkehren wollen, brauchen Sie nur `[Strg][Untbr]` zu drücken.

**Starten des
seriellen Treibers
auf dem Remote-
System**

Wenn Sie die beiden Systeme mit einem NULL-Modem-Kabel verbunden haben, müssen Sie die Kommandozeilenoptionen **-rs** und **-rp** setzen, um Geschwindigkeit und Schnittstelle für die Datenübertragung auszuwählen.

Wenn Ihre serielle Schnittstelle als COM1 konfiguriert ist, dann starten Sie TFREMOTE mit

```
TFREMOTE -rp1 -rs4
```

Das ist allerdings die Voreinstellung von TREMOTE; die Eingabe

```
TFREMOTE
```

führt zum gleichen Ergebnis, wenn die Voreinstellungen nicht geändert wurden. Wenn Ihre serielle Schnittstelle als COM2 konfiguriert wurde, wird TFREMOTE mit

```
TFREMOTE -rp2 -rs4
```

gestartet.

Alle drei Befehle starten die Verbindung mit der höchsten Geschwindigkeit (115 KBAud). Das funktioniert mit den meisten PCs und Kabeln. Wenn es Probleme mit der Kommunikation gibt, können Sie in der Tabelle C.1 anschauen, wie Geschwindigkeit verringert werden kann

Hinweis

Möglicherweise verwenden das lokale und das Remote-System für die NULL-Modem-Verbindung zwei verschiedene serielle Schnittstellen. In diesem Fall müssen Sie unterschiedliche Kommandozeilenoptionen für das lokale und das Remote-System verwenden. Außerdem müssen natürlich die Übertragungsgeschwindigkeiten beider Systeme übereinstimmen.

**Starten des LAN-
Treibers auf dem
Remote-System**

Wenn Sie die beiden Systeme mit einem LAN verbunden haben, müssen Sie TFREMOTE mit der Option **-rn** starten. Das folgende Kommando startet, wenn es auf der DOS-Ebene eingegeben wird, TFREMOTE auf einer LAN-Verbindung und gibt dem Remote-System den Namen *remotelink*:

```
TFREMOTE -rnremotelink
```

Wenn in diesem Kommando kein Remote-Name angegeben wird, bleibt der Name REMOTE aus der Voreinstellung gültig.

Näheres für die Namensvergabe bei Remote-Systemen erfahren Sie im Abschnitt „LAN-Verbindungen“ weiter hinten in diesem Anhang.

DOS-Verbindung aktivieren

Nachdem TFREMOTE auf dem Remote-System geladen wurde, starten Sie Turbo Profiler auf dem lokalen System; verwenden Sie die Kommandozeilenparameter, die der einzurichtenden Verbindung entsprechen (seriell oder LAN).

Die serielle Verbindung

Für Turbo Profiler und TFREMOTE gilt bei der Angabe der Geschwindigkeit und der Einstellung der Schnittstelle für die Datenübertragung die gleiche Syntax. Damit die Verbindung korrekt arbeiten kann, müssen beide Systeme auf die gleiche Übertragungsgeschwindigkeit eingestellt werden. Dazu dient die Kommandozeilenoption **-rs**.

Nachdem TFREMOTE auf dem Remote-System geladen wurde, vervollständigen Sie die Verbindung, indem Sie Turbo Profiler auf dem lokalen System starten. Turbo Profiler wird durch folgendes Kommando geladen, wobei eine Verbindung über den seriellen Port 2 mit einer Übertragungsgeschwindigkeit von 115000 Baud hergestellt wird:

```
TPROF -rp2 Dateiname
```

Wenn die Verbindung erfolgreich hergestellt wurde, erscheint auf dem Remote-System die Meldung *Link established*, und die Aktivitätsanzeige des lokalen Systems steht auf *Ready*. Auf dem lokalen System folgen dann die normalen Anzeigen von Turbo Profiler.

Wenn sich das Programm *Dateiname* nicht auf dem Remote-System befindet, schickt Turbo Profiler das Programm über die Verbindung auf den Remote-Rechner. Weitere Informationen über das Laden beziehungsweise Übertragen von Programmen über die Verbindung zweier Systeme finden Sie weiter unten.

Anstelle der Kommandozeilenschalter **-rs** und **-rp** kann auch **-r** verwendet werden; dadurch wird die serielle Verbindung unter Verwendung der voreingestellten Übertragungsgeschwindigkeiten und Schnittstellen aufgebaut. Wenn Sie die Voreinstellungen nicht mit dem Programm TFINST geändert haben, legt die Option **-r** COM1 auf 115000 Baud fest.

LAN-Verbindungen

TPROF benötigt den Kommandozeilenschalter **-rn**, um eine Remote-Verbindung über ein LAN aufzubauen. Bei Turbo Profiler gilt hier allerdings eine etwas andere Syntax als bei TFREMOTE. Die Turbo Profiler-Syntax für die Remote-LAN-Analyse lautet:

```
TPROF -rn [Lokal] [:Remote] Dateiname
```

Wenn bei Turbo Profiler die Kommandozeilenoption **-rn** verwendet wird, kann man zwei Parameter angeben: den Name des lokalen Systems und den des Remote-Systems; beide werden durch ein Semikolon voneinander getrennt. Da beide Parameter optionale Eingaben sind, sind bei Turbo

Profilierer vier Möglichkeiten bei der Anwendung der Kommandozeilenoption **-rn** denkbar, diese DOS-Kommandos laden alle Turbo Profiler, definieren eine Remote-LAN-Verbindung und laden das Programm *Dateiname* für die Analyse:

```
TPROF -rn Dateiname
```

Dieses Kommando verwendet sowohl für das lokale als auch für das Remote-Gerät den vorgegebenen Namen, also LOCAL und REMOTE.

```
TPROF -rnLOCAL Dateiname
```

Dieses Kommando legt LOCAL1 als den Namen des lokalen Systems fest, verwendet aber für das Remote-System dessen voreingestellten Namen (also REMOTE).

```
TPROF -rn;REMOTE1 Dateiname
```

Dieses Kommando verwendet den voreingestellten Namen für das lokale System (also LOCAL) und definiert den Namen REMOTE1 für das Remote-System.

```
TPROF -rnLocal1;Remotel Dateiname
```

Im vierten Kommando schließlich werden beide Namen neu festgelegt. (Die Namen dürfen jeweils maximal 16 Zeichen lang sein.)

Der Verbindungsaufbau sollte 15 Sekunden nach Eingabe des Aufrufs von Turbo Profiler abgeschlossen sein.

Hinweis

Wenn nur eine Person im Netzwerk Remote-Analyse betreibt, reichen auch die voreingestellten Namen für das lokale und das Remote-System; andernfalls müssen Sie allen Systemen eigene Namen geben.

Remote-Analyse von Windows-Anwendungen

Wenn Sie eine Windows-Anwendung einer Remote-Analyse unterziehen wollen, müssen Sie Windows, WREMOTE und die Anwendung auf dem einen Rechner und Turbo Profiler auf dem anderen Rechner starten. In dieser Beschreibung wird der Rechner, auf dem Windows, WREMOTE und die Anwendung laufen, als *Remote-Rechner* bezeichnet und der andere Rechner, auf dem Turbo Profiler läuft, als *lokaler Rechner*.

Einrichten des Remote-Systems

Kopieren Sie den Treiber WREMOTE.EXE, das Konfigurationsprogramm WRSETUP.EXE und alle Dateien, die das zu untersuchende Programm benötigt, auf das Remote-System. Zu diesen Dateien gehören Dateien mit Eingabedateien, Konfigurationsdateien, Hilfsdateien, DLL-Dateien für Windows und so weiter. Sie können auch Ihr Anwendungsprogramm auf

das Remote-System kopieren. Allerdings übernimmt Turbo Profiler diese Übertragung nötigenfalls automatisch.

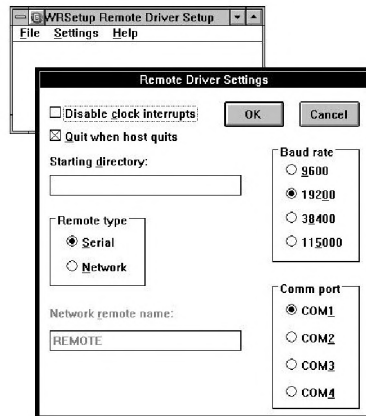
Diese Übertragungen können Sie mit Hilfe von Disketten oder auch mit dem Hilfsprogramm TDRF durchführen. Das Programm TDRF wird in den Online-Dateien von Turbo Profiler beschrieben.

Die Konfiguration von WREMOTE

Bevor Sie WREMOTE zum ersten Mal starten, führen Sie zuerst das Programm WRSETUP aus, um die Einstellungen für die Verbindung festzulegen. Wenn Sie WRSETUP zum ersten Mal starten, verwenden Sie dazu am besten das Windows *File/New*-Kommando aus dem Programm-Manager, um damit WRSETUP als Anwendung in der Windows-Anwendungsgruppe zu konfigurieren. Das gleiche müssen Sie mit WREMOTE.EXE tun.

Wenn Sie das Programm WRSETUP ausführen, sehen Sie ein Fenster mit den Kommandos *File*, *Settings* und *Help*. Wenn Sie *Settings* wählen, erhalten Sie die folgende Bildschirmdarstellung:

Abbildung C.1:
WRSETUP-
Hauptfenster und
das Dialogfenster
Settings



Serielle Konfiguration

Wenn Sie mit serieller Datenübertragung arbeiten, wählen Sie das Schaltfeld *Serial Remote Type* und geben die für Ihre Hardware passende Baud-rate und den Datenübertragungsport an. Die vorgegebenen Einstellungen hierfür sind 19200 Baud und COM1.

LAN-Konfiguration

Wenn Sie mit einem LAN arbeiten, wählen Sie das Schaltfeld *Network Remote Type* und geben im Feld *Network Remote Name* den betreffenden Remote-Namen ein. Dieser Name lautet per Vorgabe REMOTE. Weitere Informationen über die Namensvergabe an Remote-Systeme finden Sie in diesem Anhang weiter oben.

Im Eingabefeld *Starting Directory* geben Sie den Pfad des Verzeichnisses an, in dem Turbo Profiler das zu untersuchende Programm finden soll.

Wenn Sie wünschen, daß WREMOTE die Steuerung an Windows übergeben soll, nachdem die Arbeit mit Turbo Profiler auf dem lokalen System beendet wurde, wählen Sie *Quit When Host Quits*. Wenn Sie mit höheren Übertragungsgeschwindigkeiten (38400 oder 115000 Baud) arbeiten, wählen Sie *Disable Clock Interrupts*. Dann ist es für WRSETUP und Turbo Profiler leichter, Verbindungen in einer Windows-Umgebung aufzubauen.

Wie jede .INI-Datei kann auch diese mit Hilfe eines beliebigen Textverarbeitungsprogramms, das ASCII-Text produziert, editiert werden.

Wenn Sie Ihre Optionen eingegeben und das WRSETUP-Fenster geschlossen haben, sichert WRSETUP Ihre Einstellungen in der Datei WREMOTE.INI in Ihrem Windows-Verzeichnis. Die folgende WREMOTE.INI-Datei setzt WREMOTE auf 19200 Baud auf COM2, die Clock-Interrupts sind außer Kraft gesetzt und das Programm gibt nach Beendigung von Turbo Profiler die Steuerung an Windows ab:

```
[WRemote]
BaudRate=2
Port=2
Quit=1
Clock=1
```

Die Kommandozeilenoptionen von WREMOTE

Die Kommandozeilenoptionen von WREMOTE dienen dazu, die Einstellungen, die per Vorgabe gelten und die, die in der Datei WREMOTE.INI enthalten sind, zu ändern. Die Optionen beginnen entweder mit einem Bindestrich (-) oder mit einem Schrägstrich (/).

Tabelle C.2:
Die Kommandozeilenoptionen von WREMOTE

Option	Beschreibung
-cDatei	Verwende <i>Datei</i> als Konfigurations-(INI-)Datei
-dVerz	Startup-Verzeichnis <i>Verz</i> verwenden
-rc0	Clock-Interrupts eingeschaltet
-rc1	Clock Interrupts ausgeschaltet
-rnRemote-Name	Remote-Analyse mit Hilfe eines LANs
-rp1	Schnittstelle 1, (COM1); Standardeinstellung
-rp2	Schnittstelle 2, (COM2)
-rp3	Schnittstelle 3, (COM3)
-rp4	Schnittstelle 4, (COM4)
-rq0	Nicht beenden, wenn Turbo Profiler beendet wird
-rq1	Beenden, wenn Turbo Profiler beendet wird
-rs1	Langsamste Übertragungsgeschwindigkeit, 9600 Baud
-rs2	Langsame Übertragungsgeschwindigkeit, 19200 Baud
-rs3	Mittlere Übertragungsgeschwindigkeit, 38400 Baud
-rs4	Hohe Übertragungsgeschwindigkeit, 115000 Baud

Starten des Remote-Treibers für Windows

Nachdem Sie WREMOTE von Windows aus gestartet haben, stellt das Programm an der Stelle, an der sich der Mauszeiger befindet, eine Sanduhr dar; dadurch wird angezeigt, daß es bereit ist, am anderen Ende der Verbindung Turbo Profiler zu starten.

Wenn Sie WREMOTE beenden wollen, während es darauf wartet, eine Verbindung mit Turbo Profiler aufbauen zu können, drücken Sie auf dem Remote-System die Tastenkombination [Strg][Untr].

Die Remote-Verbindung zum Windows-Rechner

Wenn Sie für die Verbindung der beiden Systeme ein NULL-Modem-Kabel verwenden, können Sie die beiden Kommandozeilenoptionen **-rs** und **-rp** verwenden, um Geschwindigkeit und Übertragungsrate der Verbindung festzulegen.

Im Abschnitt „Serielle Verbindung“ etwas weiter oben finden Sie mehr Informationen über Kommandozeilen-Optionen

Turbo Profiler und WREMOTE müssen auf die gleiche Übertragungsgeschwindigkeit eingerichtet sein. Sie können entweder den Parameter **-rs** verwenden, um die Baudrate für den Turbo Profiler anzugeben oder die Kommandozeilenoption **-r**, die die serielle Remote-Verbindung aufbaut und dabei die vorgegebenen Einstellungen für Übertragungsrate und Port beibehält. Wenn Sie nicht mit Hilfe des Programms TFINST die Voreinstellungen geändert haben, wird durch die Option **-r** die Schnittstelle COM1 verwendet und die Übertragungsrate beträgt 115.000 Baud.

LAN-Verbindung

Turbo Profiler richtet eine Remote-Verbindung über ein LAN ein, wenn die Kommandozeilenoption **-rn** gesetzt wird.

Ein typischer Aufruf von Turbo Profiler für den Aufbau einer Remote-Verbindung zu einem Windows-Gerät sieht folgendermaßen aus:

```
TPROF -rs2 -w Programmname
```

Dieses Kommando baut die Verbindung über den voreingestellten seriellen Port (normalerweise COM1) auf, und die Übertragungsgeschwindigkeit beträgt 19200 Baud; das Programm *Programmname* wird auf das Remote-System übertragen, sofern es sich nicht bereits dort befindet.

Wenn Turbo Profiler auf dem lokalen Gerät gestartet wird, erscheinen die Copyright-Notiz und die folgende Meldung:

```
Waiting for Handshake from remote driver (ctrl-break to quit)
```

(Verbindungsaufbau; Abbruch mit [Strg][Untr]). Während Sie auf die Verbindung warten, erscheint ein Stundenglas auf dem Remote-System. Wenn die Verbindung hergestellt werden konnte, verschwindet das Stundenglas, und die gewohnte Bildschirmdarstellung von Turbo Profiler erscheint auf

dem lokalen Gerät. Drücken Sie `[Strg][Untbr]`, um Turbo Profiler zu verlassen, wenn die Verbindung nicht erfolgreich hergestellt werden konnte.

Das Laden von Programmen auf das Remote-System

Wenn als Kommandozeilenargument für Turbo Profiler ein Dateiname verwendet wurde, oder wenn Sie mit dem Kommando *File/Open* eine neue Datei in Turbo Profiler laden, überprüft Turbo Profiler automatisch, ob das Programm auf das Remote-System übertragen werden muß.

DLLs werden nicht automatisch auf das Remote-System übertragen.

Turbo Profiler verhält sich beim Laden des Programms auf dem *Remote-System* sehr intelligent. Er vergleicht zunächst Datum und Uhrzeit der Programmdateien des lokalen und des ferngesteuerten Systems. Wenn das Programm auf dem Remote-System nicht existiert, wird sie sofort dorthin übertragen. Wenn das Programm auf dem Remote-System existiert, überprüft Turbo Profiler Datum und Erstellungszeit des Programms auf dem lokalen Gerät und vergleicht diese Werte mit den entsprechenden Einträgen der Datei auf dem Remote-System. Wenn die lokale Kopie jüngerem Datum ist als die Kopie auf dem Remote-System, geht Turbo Profiler davon aus, daß das Programm neu kompiliert und/oder gelinkt wurde, und sendet es mit der mittels **-rs** voreingestellten Übertragungsgeschwindigkeit über die Verbindung an das Remote-System.

Bei der höchsten Übertragungsgeschwindigkeit (**-rs4**) beträgt die effektive Übertragungsrate etwa 10 KByte pro Sekunde. Das bedeutet, daß ein typisches Programm mit 60 KByte in etwa 6 Sekunden übertragen wird: kein Grund zur Beunruhigung also, wenn beim Laden eines neuen Programms eine kurze Pause auftritt.

Um anzuzeigen, daß etwas passiert, werden bei der Übertragung die übertragenen Bytes auf dem Bildschirm des Remote-Systems hochgezählt.

Der Ablauf der Remote-Analyse

Sobald Sie Turbo Profiler (mit *TFREMOTE* oder *WREMOTE*) im *Remote-Modus* gestartet haben, wird Ihr Programm so untersucht, als ob Sie nur ein einziges System hätten. Die Turbo-Profiler-Befehle arbeiten auf genau die gleiche Weise; Sie brauchen nichts Neues dazuzulernen.

Beachten Sie aber, daß alle Bildschirmausgaben oder Tastatureingaben auf dem Remote-System vor sich gehen, da das Programm, das untersucht wird, auf dem Remote-System läuft. Der Befehl *Window/User Screen* wird beim Arbeiten mit einem Remote-System nicht ausgeführt.

Im CPU-Fenster erscheint als CPU-Typ der Prozessor des ferngesteuerten Systems. Davor steht die Bezeichnung REMOTE.

Wenn Sie Dateien zum Remote-System übertragen wollen, während Sie mit Turbo Profiler arbeiten, müssen Sie DOS mit dem Befehl *File/DOS shell* aufrufen und können dann mit dem Hilfsprogramm TDRF Ihre Dateiverwaltungsoperationen auf dem ferngesteuerten System vornehmen. Mit der Eingabe von `EXIT` kehren Sie wieder zu Turbo Profiler zurück und können Ihr Programm weiter untersuchen.

Fehler und Probleme

Im folgenden finden Sie einige Methoden der Fehlersuche, die Sie ausprobieren können, wenn Sie bei der Remote-Analyse auf Schwierigkeiten stoßen:

- Überprüfen Sie Ihre Kabelverbindungen.
- Überprüfen Sie, ob Sie die richtige serielle Schnittstelle verwenden oder ob Sie korrekt an das Netzwerk angeschlossen sind.
- Versuchen Sie, die Verbindung mit einer geringeren Übertragungsgeschwindigkeit zu betreiben, bis Sie eine Geschwindigkeit ermittelt haben, bei der die Verbindung arbeitet (verwenden Sie dazu die Kommandozeilenoption **-rs**).
- Es gibt Kombinationen aus Hardware und Kabeln, die bei sehr hohen Übertragungsgeschwindigkeiten nicht zuverlässig sind. Wenn Ihre Übertragungswege bei geringeren Geschwindigkeiten funktionieren, sollten Sie sich also nach einem anderen Kabel oder einem anderen Rechner umsehen.
- Wenn Sie ein Windows-Programm untersuchen und die Verbindung bei keiner Geschwindigkeit funktioniert, verwenden Sie `WRSETUP`, um die Clock-Interrupts auszuschalten und versuchen eine Übertragungsgeschwindigkeit von 9600 Baud. Wenn das klappt, versuchen Sie es stufenweise mit höheren Geschwindigkeiten.

Meldungen von TFREMOTE

nn bytes downloaded

[nn Bytes übertragen]

Eine Datei wurde zum *Remote*-System übertragen. Diese Mitteilung gibt den Fortgang der Dateiübertragung wieder. Bei der höchsten Übertragungsrate (115.000 Baud) werden ungefähr 10 KB pro Sekunde übertragen.

Can't create file

[Datei kann nicht angelegt werden]

Sie versuchen ein Programm zu untersuchen und die Datei existiert nicht auf dem *Remote*-System. TFREMOTE hat dies überprüft und festgestellt, daß es eine Datei, die übertragen werden soll, nicht anlegen kann. Das passiert entweder, wenn das Laufwerk voll ist, oder wenn der Dateiname bereits als Verzeichnis existiert.

Can't modify exe file

[.EXE-Datei kann nicht verändert werden]

Der für die Modifikation angegebene Dateiname enthält keine gültige Kopie von TFREMOTE. Sie können nur Kopien von TFREMOTE mit der Option **-w** modifizieren.

Can't open exe file to modify

[.EXE-Datei kann nicht für eine Änderung geöffnet werden]

Die Datei, die verändert werden soll, kann nicht geöffnet werden. Sie haben wahrscheinlich einen ungültigen Dateinamen angegeben.

Download complete

[Übertragung abgeschlossen]

Die Datei wurde korrekt an TFREMOTE übertragen.

Download failed, write error on disk

[Übertragung nicht geglückt, wegen Schreibfehler auf Laufwerk]

TFREMOTE kann einen Teil der empfangenen Datei nicht auf das Laufwerk schreiben. Das passiert normalerweise dann, wenn das Laufwerk voll ist. Sie müssen einige Dateien löschen, bevor die Datei übertragen werden kann.

Enter program file name to modify

[Geben Sie den Namen der Programmdatei ein, die modifiziert werden soll]

Wenn Sie mit MS-DOS 3.0 oder einer der nachfolgenden Versionen arbeiten, enthält die Eingabeaufforderung bereits den Pfad und den Namen der Programmdatei, mit der TFREMOTE gestartet wurde. Sie können diesen Namen mit [Eingabe] akzeptieren oder eine andere Programmdatei angeben. Wenn Sie mit einer MS-DOS Version 2.x arbeiten, dann müssen Sie den Pfad und den Namen der Programmdatei vollständig angeben.

Interrupted

[Unterbrochen]

Sie haben [Strg][Untbr] gedrückt, während Sie auf das Zustandekommen der Verbindung gewartet haben.

Invalid command line option

[Unzulässige Kommandozeilenoption]

Sie haben eine ungültige Kommandozeilenoption beim Start von TDRF aus einer DOS-Befehlszeile eingegeben.

Link broken

[Verbindung unterbrochen]

Das Programm, das mit TFREMOTE in Verbindung war, wurde abgebrochen und ist zu MS-DOS zurückgekehrt.

Link established

[Verbindung zustandegekommen]

Ein Programm auf dem anderen System hat gerade die Verbindung mit TFREMOTE aufgenommen.

Loading program *name* from disk

[Programm *Name* wird vom Laufwerk geladen]

Turbo Profiler hat TFREMOTE mitgeteilt, daß ein Programm zum Untersuchen vom Laufwerk in den Speicher geladen werden soll.

No network present

[Kein Netzwerk gefunden]

TFREMOTE findet kein NETBIOS-kompatibles Netzwerk. Stellen Sie sicher, daß Sie NETBIOS (Version 3.0 oder später) geladen haben und daß die Verbindung zum Netzwerk »steht«.

Program load failed, EXEC failure

[Programm kann nicht geladen werden, EXEC-Fehler]

MS-DOS konnte das Programm nicht in den Speicher laden. Das tritt dann auf, wenn das Programm beschädigt oder abgeschnitten ist. Sie sollten das Programm vom Laufwerk löschen. Damit wird Turbo Profiler veranlaßt, eine neue Kopie über die Verbindung zu übertragen. Wenn diese Meldung auch nach dem Löschen der Datei wieder auftritt, sollten Sie die Verbindung neu aufbauen und es noch einmal versuchen.

Program load failed, not enough memory

[Programm kann nicht geladen werden, da zuwenig Speicherplatz frei ist]

Das Remote-System hat nicht genug freien Speicherplatz, um das Programm zu laden, das untersucht werden soll.

Program load failed; program not found

[Programm kann nicht geladen werden, da das Programm nicht gefunden werden konnte]

TFREMOTE hat das Programm auf dem Laufwerk nicht gefunden. Das sollte niemals passieren, da Turbo Profiler das Programm auf das Remote-System überträgt, wenn die Datei nicht vorhanden ist.

Program load successful

[Programm korrekt geladen]

TFREMOTE hat das Programm korrekt geladen, das mit Turbo Profiler debuggt werden soll.

Reading file *name* from Turbo Profiler

[Datei *Name* wird gerade von Turbo Profiler gelesen]
Datei wird gerade an Turbo Profiler übertragen.

Unknown request: *message*

[Unbekannte Anforderung: *Meldung*]

hat eine unzulässige Anforderung von dem anderen System erhalten. Diese Meldung sollte niemals auftreten, wenn die Verbindung in Ordnung ist. Überprüfen Sie bitte das Verbindungskabel. Wenn Sie diese Meldung weiterhin erhalten, sollten Sie versuchen, die Übertragungsgeschwindigkeit mit der Kommandozeilenoption **-rs** zu reduzieren.

Waiting for handshake (press Ctrl-Break to quit)

[Warte auf Handshake-Signal (drücken Sie [Strg][Untbr], um abzubrechen)]
TDREMOTE wurde gestartet und wartet auf ein Programm, das die Verbindung vom anderen System her aufnimmt. Wenn Sie zu MS-DOS zurückkehren wollen, bevor das andere System die Kommunikation anfängt, müssen Sie die Tastenkombination [Strg][Untbr] drücken.

WREMOTE-Meldungen

Can't find configuration file

[Die Konfigurationsdatei kann nicht gefunden werden]
Sie haben mit der Kommandozeilenoption **-c** eine Datei angegeben, die nicht existiert.

Can't load WINDEBUG.DLL

[WINDEBUG.DLL kann nicht geladen werden]
Die dynamische Link-Bibliothek WINDEBUG.DLL befindet sich nicht im aktuellen Verzeichnis. Diese Datei ist für den Ablauf von WREMOTE notwendig.

Can't open COMx serial port

[Die serielle Schnittstelle COMx kann nicht geöffnet werden]
WREMOTE versucht eine Schnittstelle zu verwenden, die entweder belegt ist oder nicht existiert.

Invalid switch

[Ungültiger Schalter]

Sie haben in der Kommandozeile von WREMOTE eine unbekannte Option angegeben.

No network present

[Kein Netzwerk gefunden]

WREMOTE findet kein NETBIOS-kompatibles Netzwerk. Überzeugen Sie sich davon, daß Sie NETBIOS (Version 3.0 oder höher) geladen haben und ins Netzwerk eingeloggt sind.

Turbo Profiler für Windows

Mit Turbo Profiler für Windows (TPROFW) können Sie Anwendungen analysieren, die Sie für Microsoft Windows 3.0 oder eine spätere Version geschrieben haben. Das Programm läuft unter Windows auf den gleichen Geräten wie das Programm, das Sie analysieren, und es kann zwischen der TPROFW-Bildschirmdarstellung und der Bildschirmdarstellung Ihres Anwendungsprogramms umgeschaltet werden.

Die Analyse verläuft unter Windows größtenteils so wie unter DOS, außer daß Sie auch auf Informationen zugreifen können, die für Anwendungen unter Windows typisch sind. Dazu gehören

- Meldungen, die von Ihrem Anwendungsprogramm gesendet und empfangen werden,
- eine vollständige Liste der Module, die von Windows geladen werden (darunter auch DLLs) und
- die Analyse von DLLs.

TPROFW läuft unter betreibt Windows im normalen oder im erweiterten 386-Modus, was bedeutet, daß Ihr Computer über einen 80386-Prozessor (oder Nachfolgeprozessor) und über mindestens 1 MB Speicher verfügen muß.

TPROFW unterstützt verschiedene Graphikkarten durch Verwendung von DLL-Dateien. Nach dem Sie TPROFW installiert haben, starten Sie TDWINI.EXE, mit dessen Hilfe Sie einen Treiber auswählen oder den eingestellten Treiber ändern können.

Standardmäßig verwendet TPROF den SVGA.DLL Graphiktreiber, der die meisten Graphikkarten und Monitore unterstützt. Weitere Informationen über verfügbare Graphik-DLLs finden Sie unter DUAL8514.DLL, STB.DLL, SVGA.DLL und TWGUI.DLL in der Online-Hilfe von TDWINI.EXE.

TPROFW kann (wie auch Turbo Profiler) einen zweiten, ebenfalls an Ihren Computer angeschlossenen Bildschirm nutzen; Sie können dann TPROFW-Bildschirmdarstellungen auf dem einen Monitor und die Darstellungen Ihrer Anwendersoftware auf dem anderen Bildschirm verfolgen. Sie wählen diese Darstellungsoption, indem Sie den Kommandozeilenschalter **-do** setzen oder das Hilfsprogramm TINST starten und bei *User Screen Updating* die Option *Other Display* auswählen.

Die Installation von TPROFW

Wenn Sie Turbo Profiler auf Ihrem System installieren, dann richtet das Installationsprogramm im gleichen Verzeichnis, in dem sich auch Ihre Turbo Profiler-Dateien befinden, die folgenden beiden Dateien ein:

- TPROFW.EXE, das TPROFW-Programm
- TFWHELP.TFH, die TPROFW-Hilfdatei

Das Installationsprogramm legt auf Ihrem System für TPROFW ein Programmsymbol an und installiert es innerhalb des Bereichs, den der Windows Programm-Manager für Ihre Borland Sprachen angelegt hat. Anschließend können Sie TPROFW durch Anklicken dieses Programmsymbol starten, wie jede andere Windows-Anwendung auch.

Installation der Datei TDDE- BUG.386

Die Datei TDDEBUG.EXE auf Ihren Installationsdisketten stellt den gleichen Funktionsumfang zur Verfügung wie die Windows-SDK-Datei WINDEBUG.386, bietet darüber hinaus jedoch eine bessere Unterstützung der Tastenkombination **[Strg][Alt][S-Abf]**. (Damit wird eine Windows-Anwendung unterbrochen und man kehrt zu TPROFW zurück).

Das Installationsprogramm sollte diese Datei auf Ihre Festplatte kopieren und die zu Windows gehörende Datei SYSTEM.INI so verändern, daß Windows anstelle der Datei WINDEBUG.386 die Datei TDDEBUG.386 lädt. Wenn das Installationsprogramm diese Aufgabe aus irgendwelchen Gründen nicht schafft, müssen Sie sich selbst darum kümmern:

1. Das Installationsprogramm hat die Datei TDDEBUG.386 von der Installationsdiskette auf Ihre Festplatte kopiert. Das Standardverzeichnis für diese Datei ist C:\TPROF\BIN. Wenn Sie diese Datei in ein anderes Verzeichnis verlagern, müssen Sie diese Änderung in den entsprechenden Anweisungen berücksichtigen.
2. Öffnen Sie mit einem Editor die Windows-Datei SYSTEM.INI, suchen Sie den Eintrag **[386enh]** und fügen Sie in dem Abschnitt mit dem Eintrag **386enh** folgende Zeile hinzu:

```
device=c:\tprof\bin\tddebug.386
```

3. Wenn es in dem Abschnitt *386enh* eine Zeile gibt, die die Datei WINDEBUG.386 lädt, müssen Sie diese Zeile mit Hilfe eines Strichpunkts auskommentieren oder ganz löschen. (Es dürfen auf keinen Fall TDDEBUG.386 und WINDEBUG.386 gleichzeitig geladen werden.)

Wenn Sie zum Beispiel WINDEBUG.386 aus dem Verzeichnis C:\WINDOWS laden, lautet die Zeile, die auskommentiert oder gelöscht werden muß:

```
;device=c:\windows\windebug.386
```

Das Konfigurieren von TPROFW

Für das Programm TPROFW gibt es – wie auch beim Turbo Profiler – zwei Methoden der Konfiguration. Eine davon ist der Einsatz von Kommandozeilenoptionen unter Verwendung von TFINST.

Der Kommandozeilenaufwurf von TPROF

TPROFW kann durch Verwendung von Kommandozeilenoptionen konfiguriert werden, auf die wahlweise noch ein Programmname mit eigenen Kommandozeilenoptionen folgen darf. Dem Programmnamen darf noch ein Pfadname vorangestellt werden. TPROFW kann durch Verwendung von Kommandozeilenoptionen konfiguriert werden, auf die wahlweise noch ein Programmname (inklusive Pfad) mit eigenen Optionen folgen kann.

Da TPROFW ein Windows-Programm ist, werden Sie die Kommandozeilenoptionen vermutlich unter Verwendung der Kommandos *File/Run* oder *File/Properties* (beide aus dem Program Manager) eingeben, um die Kommandozeileneigenschaften des TPROFW-Icons zu ändern. Sie können auch Windows starten und TPROFW von der DOS-Kommandozeile aus aufrufen. Geben Sie zuerst das Windows- und dann das TPROFW-Kommando ein; wahlweise dürfen darauf noch Schalter und ein Programmname (mit oder ohne eigene Schalter) folgen. Die Kommandozeilensyntax für TPROFW lautet:

```
TPROFW [Optionen] [Programmname [Programmarginente]]
```

In der Tabelle D.1 finden Sie eine Zusammenfassung der Kommandozeilenoptionen von TPROFW:

Tabelle D.1:
Die Kommandozeilenoptionen des Programms TPROF

Option	Wirkung
-? , -h	Gibt die erläuternden Texte zu den Kommandozeilenoptionen von TPROFW aus.
-b	Analysen werden im Stapelbetrieb durchgeführt.
-bAnzahl	Im Stapelbetrieb <i>Anzahl</i> Programmdurchläufe durchführen.
-cDatei	Verwendet die Konfigurationsdatei <i>Dateiname</i> .

-do	TPROFW läuft auf dem zweiten Bildschirm.
ds	Bildschirm-Updates werden durch Umschalten der Bildschirmseiten erzeugt.
-p	Es wird eine Maus verwendet. Wenn der Maustreiber für Windows nicht funktioniert, funktioniert er auch für TPROFW nicht; die Kommandozeilenoption hat dann keine Wirkung.
-sc	Bei Symbolnamen wird nicht zwischen Groß- und Kleinschreibung unterschieden.
-sdVerz[;Verz...]	Setzt ein oder mehrere Quelldateiverzeichnisse.
-tVerz	Setzt das Startverzeichnis.

Eine vollständige Beschreibung dieser Kommandozeilenoptionen finden Sie im Anhang A.

Hinweis

Die Kommandozeilenoption *-t* ist nur bei TPROFW verfügbar. Mit dieser Option ändert man das Startverzeichnis von TPROFW (also das Verzeichnis, in dem TPROFW die Konfigurationsdatei und die .EXE-Dateien sucht, bei denen kein vollständiger Pfad angegeben wurde). Die Syntax lautet

-tVerzeichnisname

Es kann nur ein Startverzeichnis angegeben werden. Wenn Sie dieses Kommando mehr als einmal in derselben Kommandozeile eingeben, verwendet TPROF nur den letzten Eintrag.

Die Verwendung von TFINST im Zusammenhang mit TPROFW

Um TFINST auf TPROFW anzuwenden, starten Sie TFINST zusammen mit der Kommandozeilenoption *-w*. TFINST wirkt auf TPROFW genauso wie auf Turbo Profiler, außer daß der Name der Standard-Konfigurationsdatei TFCONFIG.TWF lautet und weniger Optionen zur Verfügung stehen. Vergleichen Sie hierzu die Kommandozeilenoptionen zu TPROFW im vorhergehenden Abschnitt.

Eine Beschreibung der Anwendung von TFINST finden Sie in Anhang B.

TPROFW im Einsatz

Nach dem Laden erscheint TPROFW nicht etwa in einem Fenster; Sie haben vielmehr den normalen, zeichenorientierten DOS-Bildschirm in voller Größe vor sich. Anders als bei anderen Anwendungen, die unter Windows laufen, können Sie die Tastenkombinationen (etwa *[Alt][Esc]* oder *[Strg][Esc]*) nicht dazu verwenden, TPROFW zeitweise zu verlassen und andere Anwendungen zu starten. Wenn die Anwendung, die Sie analysieren, jedoch aktiv ist (d. h. der Cursor ist in einem der Fenster des Anwendungsprogramm aktiv) können Sie die Tastenkombinationen oder die Maus dazu verwenden, zu anderen Programmen umzuschalten.

Hinweis

Wenn Sie die Tastenkombination **[Strg][Esc]** dazu verwenden, ein Anwendungsprogramm zu verlassen, das unter TPROFW läuft, finden Sie den Namen des Anwendungsprogramms in der Task-Liste. TPROF selbst erscheint nicht in der Task-Liste, da es sich bei TPROFW nicht um eine normale Windows-Task handelt, die Sie ein- oder ausschalten können.

Die Analyse von Programmen unter TPROFW verläuft ganz ähnlich wie unter Verwendung von Turbo Profiler. Es gibt jedoch einige Unterschiede:

- Das Umschalten von Ihrem Anwendungsprogramm zu TPROFW geschieht mit Hilfe der Tastenkombination **[Strg][Alt][S-Abf]**, also analog zu der Methode, mit Hilfe der Tastenkombination **[Strg][Untbr]** eine DOS-Anwendung zu verlassen und zum Turbo Profiler zurückzukehren; der Unterschied besteht darin, daß die DOS-Anwendung dabei beendet wird, während die Windows-Anwendung nur vorübergehend verlassen wird.
- Falls möglich, sollten Sie Ihr Anwendungsprogramm bis zum Ende durchlaufen lassen oder es mit dem System-Kommando verlassen, bevor Sie TPROF beenden oder ein anderes Programm laden, um es zu analysieren. Wenn Sie vergessen, ein Windows-Programm korrekt zu verlassen, bleiben dadurch unter Umständen Speicherstellen belegt, die andernfalls freigegeben würden, wodurch Probleme mit TPROFW oder anderen Anwendungen entstehen können.
- Das DOS-Shell-Kommando steht vom *File*-Menü aus nicht zur Verfügung, da TPROFW nicht unter DOS läuft.
- Die *Edit*-Befehle der lokalen Menüs *Module* und *Text File* stehen nicht zur Verfügung.
- Die Fenster *Interrupts*, *File I/O* und *Overlay* stehen nicht zur Verfügung.
- Die Einstellungen *Display Swapping* stehen im Dialogfenster *Display Options* nicht zur Verfügung.

**Die Analyse von
Windows-Proze-
duren**

Das Analysieren von Routinen innerhalb von Windows-Anwendungen geschieht unter TPROFW auf folgende Weise: Es werden die Meldungsklassen durchgearbeitet, die durch die Routine aufgerufen werden und auch die Windows-Meldungen, die an die Prozedur gesandt werden. Bevor aber Meldungsklassen und Windows-Meldungen verfolgt werden können, müssen Sie dafür sorgen, daß es eine Bereichsmarkierung gibt, die eine Windows-Prozedur repräsentiert. Das Einrichten einer Bereichsmarkierung für eine Windows-Prozedur wird entweder vom Kommando *Operation* des lokalen Menüs aus dem Module-Fenster erledigt oder vom Kommando *Option* des lokalen Menüs aus dem Area-Fenster.

**Das Dialogfenster
Window Procedure
Messages**

Nachdem eine Bereichsmarkierung als Markierung für eine Windows-Prozedur eingerichtet wurde, müssen Sie festlegen, welche Meldungen

aus welchen Meldungsklassen Sie in dieser speziellen Prozedur ansehen wollen. Wenn *Messages* gewählt wurde

Mit dem Kommando *Operation* des lokalen Menüs aus dem *Module*-Fenster aus oder mit dem Kommando *Option* des lokalen Menüs aus dem *Area*-Fenster aus, erscheint das Eingabefenster *Window Procedure Message*.

Im Eingabefenster *Window Procedure Messages* können Sie die Windows-Meldungen und die Meldungsklassen festlegen, die im Rahmen der gerade geltenden Bereichsmarkierung durchgesehen werden sollen. Die Voreinstellung veranlaßt TPROFW dazu, alle Meldungsklassen durchzuarbeiten. Die Text- und Eingabefelder des Dialogfensters *Windows Procedure Messages* haben folgende Aufgaben:

- Unter *Windows Messages* finden Sie spezielle Windows-Meldungen.
- *Message Name* nimmt eine Windoss-Meldung oder die Nummer einer Windows-Meldung auf. Verwenden Sie das Kommando *Add*, um diese Meldung in die Liste aufzunehmen, die sich im Feld *Windows Message* befindet.

Turbo Profiler nimmt für Windows-Meldungen nur Bezeichnungen zur Kenntnis, die mit der Zeichenfolge *WM_* beginnen. Wenn Sie eine Meldung verfolgen möchten, die nicht mit dieser Zeichenfolge beginnt, müssen Sie mit der Nummer der entsprechenden Meldung arbeiten. Die Kennnummern der Windows-Meldungen beziehen Sie entweder aus den Quelldateien des Programms oder aus den Windows-Headerdateien, die per *INCLUDE*-Anweisung in die Quelldateien aufgenommen wurden.

Bei den Namen für die Windows-Meldungen wird zwischen Groß- und Kleinschreibung unterschieden.

- Das Kommando *Delete* löscht die gerade hell hinterlegte Meldung aus dem Feld *Windows Messages*.
- Das Kommando *Remove All* löscht alle Meldungen, die im Feld *Windows Messages* aufgeführt werden; das Feld ist anschließend leer.
- Das Kommando *Add All* wählt alle *WM_*-Meldungen aus allen Klassen aus; sämtliche Meldungen werden in das Feld *Windows Messages* aufgenommen. Da sehr viele Meldungen übermittelt werden, ist es sinnvoll, den Bereich der zu untersuchenden Meldungen einzuschränken; beschränken Sie sich deshalb bei den Meldungsnamen auf angehörige von Klassen, die für Sie von Interesse sind.

Hinweis

Bevor Sie sich für *Add All* entscheiden, müssen Sie sicherstellen, daß die Einstellung für die maximal zulässige Anzahl an Meldungen (*Statistics/Profiling Options*) auf die eventuell sehr hohe Anzahl an Meldungen eingerichtet ist, die Sie analysieren möchten.

Es gibt insgesamt über 140 verschiedene WM_-Meldungen.

- Das Kommando *Add* veranlaßt, daß der Name der Meldung, den Sie im Feld *Message Name* angegeben haben, in die Liste mit den Windows-Meldungen aufgenommen wird.
- Im Auswahlfeld *Message Classes* wählen Sie die Meldungsklassen aus, die Sie überwachen wollen. Wenn Sie eine bestimmte Meldungsklasse auswählen, werden alle WM_-Meldungen, die zu dieser Klasse gehören, verfolgt.

Tabelle D.2:
Die Klassen der
Windows-Meldungen

Meldungsklasse	Beschreibung
All Messages	Alle mit WM_ beginnenden Meldungen.
Mouse	Meldungen, die durch eine Maus-Aktion erzeugt wurden (z. B. WM_LBUTTONDOWN und WM_MOUSEMOVE).
Window	Meldungen vom Window-Manager (z. B. WM_Paint und WM_CREATE).
Input	Meldungen, die durch eine Aktion mit der Tastatur oder durch einen Anwenderzugriff auf ein Systemmenü, einen Rollbalken oder eine <i>Size Box</i> ausgelöst werden (z. B. WM_KEYDOWN).
System	Meldungen, die aufgrund einer systemumfassenden Änderung entstehen (z. B. WM_FONTCHANGE und WM_SPOOLERSTATUS).
Initialization	Meldungen, die erzeugt werden, wenn eine Anwendung ein Eingabefeld oder ein Fenster ausgibt (z. B. WM_INITDIALOG und WM_MENU).
Clipboard	Meldungen, die ausgegeben werden, wenn eine Anwendung einen Zugriff auf ein Fenster in einer anderen Anwendung Fenster versucht (z. B. WM_DRAWCLIPBOARD und WM_SIZECLIPBOARD).
DDE	Meldungen im Zusammenhang mit dem dynamischen Datenaustausch, die durch die Kommunikation eines Anwenderprogramms mit den Fenstern eines anderen Anwendungsprogramms verursacht werden (z. B. WM_DDE_INITIATE und WM_DDE_ACK).
Non-client	Meldungen, die Windows für einen Bereich der Art <i>Non-Client</i> für ein Anwendungsfenster erzeugt.
Other	Jede Meldung, die mit WM_ beginnt und in keine der anderen Kategorien fällt

Eine vollständige Liste aller WM_Meldungen finden Sie in der Online-Hilfe des Borland Compilers.

Hinweis

Nachdem Sie die passenden Windows-Meldungen zusammengestellt haben, analysieren Sie Ihr Programm wie gewohnt. Um sich die Statistiken über die Windows-Meldungen anzusehen, wählen sie im lokalen Menü von Execution Profile im Dialogfenster *Display Options* den Schalter *Windows Procedures*.

Die Analyse von DLLs

Eine DLL (*Dynamic Link Library*) ist eine Windows-Bibliothek von Routinen und Ressourcen, die nicht während der Compilierung, sondern wäh-

rend der Laufzeit in Ihre Anwendung eingebunden wird. Dieses „Linken während der Laufzeit“ macht es möglich, daß eine Vielzahl von Anwendungen mit einer einzigen Kopie von Routinen, Daten oder Gerätetreibern arbeiten kann. Wenn eine Anwendung gestartet wird, die mit DLLs arbeitet, lädt Windows diese DLLs (falls es nicht schon vorher geschehen ist) in den Speicher, so daß das Anwendungsprogramm einen Zugriff auf den Einsprungspunkt zur DLL hat.

TPROFW kann auch eine DLL laden, die nicht über eine Symboltabelle verfügt; jedoch nur in das CPU-Fenster.

Wenn Sie ein Anwendungsprogramm laden, daß mit DLLs verbunden (gelinkt“) ist, stellt TPROFW fest, welche dieser DLLs, wenn überhaupt welche, über Symboltabellen verfügen (d. h. mit der entsprechenden Debug-Option kompiliert wurden) und arbeitet diese DLLs entsprechend ab.

TPROFW lädt automatisch die Symboltabelle und die Quelldatei einer jeden DLL, die mit Ihrem Anwendungsprogramm gelinkt ist – aber nur, wenn diese DLL über eine kompatible Symboltabelle verfügt. Eine DLL hat nur dann eine zu TPROFW kompatible Symboltabelle, wenn bei der Kompilierung die Profile-Informationen eingeschaltet waren; als Compiler muß Borland C++ oder Pascal für Windows verwendet worden sein.

Hinweis

DLLs, die mit Hilfe des Aufrufs *LoadLibrary* geladen wurden, werden nicht automatisch von Turbo Profiler analysiert. Um diese DLLs zu analysieren, müssen Sie nach dem *LoadLibrary*-Aufruf eine Bereichsende-Markierung setzen. Wenn der Profiler auf diese Markierung trifft, steht Ihnen über das Kommando *Module* im lokalen Menüs des Module-Fensters oder über *View/Module* Zugriff auf die DLLs.

Da die Symboltabelle für die DLL mit der Symboltabelle für das ausführbare Programm nicht verbunden ist, erzeugt Turbo Profiler für jede analysierte DLL eine eigene Statistikdatei (.TFS) und eine eigene Bereichsdatei (.TFA) .

Fehlermeldungen unter TPROFW

Es gibt nur eine einzige Fehlermeldung, die ausschließlich von TPROFW ausgegeben wird. Darüber hinaus können auch Fehlermeldungen von Turbo Profiler ausgegeben werden:

Ctrl-Alt-SysRq interrupt. System crash possible.Continue?

[Strg][Alt][S-Abf]-Interrupt. Systemabsturz möglich. Weitermachen?

Sie haben zuerst mit Hilfe der Tastenkombination [Strg][Alt][S-Abf] Ihr Anwendungsprogramm unterbrochen und anschließend versucht, TPROFW

zu verlassen oder Ihre Anwendung nochmals zu laden. Da der Windows-Kernel aktiv war, während Sie das Anwendungsprogramm unterbrochen haben, hat das Verlassen von TPROFW oder das erneute Laden des Anwendungsprogramms unvorhersehbare Resultate zur Folge, sehr wahrscheinlich einen Systemabsturz, der einen Neustart erzwingt.

Eingabeaufforderungen und Fehlermeldungen

Turbo Profiler gibt Meldungen und Eingabeaufforderungen an der aktuellen Cursor-Position aus. Dieses Kapitel beschreibt die Eingabeaufforderungen so wie die Fehler- und Informationsmeldungen, die von Turbo Profiler erzeugt werden.

Hier wird erklärt, wie sie auf Eingabeaufforderungen und Fehlermeldungen reagieren sollten. Alle Eingabeaufforderungen und Fehlermeldungen sind in alphabetischer Reihenfolge aufgeführt und werden einzeln beschrieben.

Eingabeaufforderungen in Turbo Profiler

Turbo Profiler gibt eine Eingabeaufforderung in einem Dialogfenster aus, wenn zusätzliche Informationen benötigt werden, um einen Befehl abzuschließen. Die Eingabeaufforderung beschreibt die benötigten Informationen. Als Inhalt kann eine History-Liste mit den früher bereits gegebenen Antworten erscheinen.

Sie haben zwei Möglichkeiten, um auf eine Eingabeaufforderung zu antworten:

- Geben Sie die Antwort ein und drücken Sie [*Eingabe*].
- Drücken Sie [*Esc*], um den Dialog abubrechen und zu dem Menübefehl zurückzukehren, mit dem der Dialog eingeleitet wurde.

Einige Eingabeaufforderungen lassen Ihnen nur die Wahl zwischen zwei Entscheidungen (wie *Yes/No*). Mit Hilfe von [*Tab*] können Sie die ge-

wünschte Auswahl treffen und mit [*Eingabe*] bestätigen, oder [*Y*] oder [*N*] direkt eingeben. Abbrechen läßt sich der Befehl mit [*Esc*].

Eine ausführlichere Behandlung der Tastaturbefehle eines Dialogfensters finden Sie in Kapitel 2.

Hier nun eine alphabetisch sortierte Zusammenfassung aller Eingabeaufforderungen und Meldungen, die in Dialogfenstern vorkommen können:

Enter code label to position to

[Geben Sie die Code-Position an, zu der Sie springen wollen]

Geben Sie die Adresse ein, die Sie im Code-Ausschnitt untersuchen wollen. Der Code-Ausschnitt zeigt die disassemblierten Befehle ab der angegebenen Adresse.

Enter command line arguments

[Geben Sie Kommandozeilenargumente an]

Geben Sie die Kommandozeilenargumente für das Programm an, das Sie untersuchen. Sie können die aktuellen Kommandozeilenargumente akzeptieren oder neue eingeben.

Anschließend werden Sie gefragt, ob das Programm erneut geladen werden soll. Bei einigen Programmiersprachen oder beispielsweise Programmen, die in C geschrieben sind, ist es erforderlich, das Programm neu zu laden, bevor die Argumente berücksichtigt werden.

Enter file name to restore from

[Geben Sie den Namen der Datei an, aus der die Statistiken geladen werden sollen]

Geben Sie den Namen der Datei an, aus der die Statistiken geladen werden sollen. Wenn Sie eine Erweiterung für den Dateinamen angeben, wird diese verwendet. Sonst wird die Dateierweiterung ».TFS« verwendet.

Enter file name to save areas to

[Geben Sie den Namen der Datei ein, in der die Informationen über die Bereichsmarkierungen gespeichert werden sollen]

Geben Sie den Namen der Datei ein, in der die aktuellen Bereichsmarkierungen gespeichert werden sollen. Wenn Sie eine Erweiterung für den Dateinamen angeben, wird diese verwendet. Sonst wird die Dateierweiterung .TFA angehängt.

Enter file name to save to

[Geben Sie den Namen der Datei zum Speichern ein]

Geben Sie den Namen der Datei ein, in der die aktuellen Statistiken gespeichert werden sollen. Wenn Sie eine Erweiterung für den Dateinamen angeben, wird diese verwendet. Sonst wird die Dateierweiterung .TFS angehängt.

Enter Interrupt number

[Geben Sie die Nummer des Interrupts ein]

Geben Sie den Namen des Interrupts ein, der protokolliert werden soll.

Enter name of file to view

[Geben Sie den Namen der Datei ein, die Sie anzeigen wollen]

Geben Sie den Namen einer Textdatei ein, die Sie untersuchen wollen.

Die angegebene Datei erscheint im File-Fenster.

Enter new directory

[Geben Sie das neue Verzeichnis an]

Geben Sie den Namen des neuen Laufwerks und/oder Verzeichnisses an, das zum aktuellen Laufwerk oder Verzeichnis werden soll.

Enter new line number

[Geben Sie die Nummer der neuen Zeile an]

Geben Sie die Nummer der neuen Zeile in der Text-Datei an, auf die der Cursor positioniert werden soll. Die erste Zeile in einer Datei ist die Zeile 1. Wenn Sie eine Zeilennummer angeben, die größer als die letzte Zeile der Datei ist, wird der Cursor in die letzte Zeile positioniert.

Enter program name to load

[Geben Sie den Namen des Programms ein, das geladen werden soll]

Geben Sie den Namen des Programms ein, das sie laden wollen. Wenn das Programm die Dateierweiterung .EXE hat, braucht diese nicht mit angegeben zu werden. Falls das Programm eine andere Erweiterung hat, müssen Sie diese Erweiterung explizit angeben.

Wenn Sie Jokerzeichen angeben oder die Standard-Vorgabe *.EXE übernehmen, erscheint eine Liste mit den übereinstimmenden Dateien, aus denen Sie die gewünschte Datei auswählen können.

Enter routine name to add

[Geben Sie den Namen der Routine ein, die hinzugefügt werden soll]

Geben Sie den Namen der Funktion ein, die sie einfügen, entfernen oder einstellen wollen.

Enter search string

[Geben Sie einen Suchstring ein]

Geben Sie die gesuchte Zeichenkombination ein. Diese kann auch Jokerzeichen enthalten, wenn Sie sich des gesuchten Strings nicht ganz sicher sind; * steht für null oder mehrere beliebige Zeichen, während ? ein einzelnes beliebiges Zeichen repräsentiert.

Enter source directory list

[Liste für Quellverzeichnis eingeben]

Geben Sie das Verzeichnis oder die Verzeichnisse an, in denen nach Quelldateien gesucht werden soll.

Wenn Sie mehr als ein Verzeichnis eingeben wollen, trennen Sie die unterschiedlichen Verzeichnispfade durch ein Leerzeichen oder ein Strichpunkt (;). Diese Verzeichnisse werden in der Reihenfolge dieser Liste nach Quelldateien durchsucht.

Pick a caller

[Wählen Sie eine der aufrufenden Routinen]

Wählen Sie eine Routine aus der Liste der aufrufenden Programmteile aus. Daraufhin wird das Fenster aktiviert, das Sie im vorigen Menü ausgewählt haben und die gewünschte Routine dargestellt.

Pick a method name

[Wählen Sie eine Methode aus]

Sie haben eine Routine angegeben, die sich auf mehr als eine Methode in einem Objekt bezieht. Wählen Sie eine Methode und die gewünschten Argumenten aus der angegebenen Liste.

Pick a module

[Wählen Sie ein Modul aus]

Wählen Sie den Namen des Moduls, das im *Module*-Fenster angezeigt werden soll. Sie erhalten eine Liste mit allen Modulen Ihres Programms. Verwenden Sie entweder die Pfeiltasten, um das gewünschte Modul zu selektieren, oder geben Sie die Anfangsbuchstaben des Moduls ein. Während der Eingabe des Modulnamens bewegt sich der Auswahlbalken auf das jeweils erste Modul, das mit den eingegebenen Buchstaben übereinstimmt. Wenn der Auswahlbalken das gewünschte Modul markiert, drücken Sie [*Eingabe*].

Pick a source file

[Wählen Sie eine Quelldatei aus]

Wählen Sie eine neue Quelldatei aus, die im *Module*-Fenster angezeigt werden soll. Die Liste enthält alle Quelldateien, aus denen sich das Modul zusammensetzt.

Pick interrupt

[Wählen Sie einen Interrupt aus]

Wählen Sie einen Interrupt aus der Liste der Interrupts, die in Turbo Profiler berücksichtigt werden.

Pick macro to delete

[Wählen Sie zu löschendes Makro aus]

Wählen Sie aus der Liste der definierten Makros dasjenige aus, das Sie löschen wollen.

Reload program and change profiling mode?

[Programm neu laden und Analyse-Modus wechseln]

Wenn Sie in Turbo Profiler vom aktiven oder passiven auf *Coverage*-Analyse umschalten, oder von *Coverage*-Analyse auf aktive oder passive Analy-

se, werden alle Statistiken gelöscht. Wenn Sie die Statistiken sichern wollen, antworten Sie bei dieser Eingabeforderung *No* und speichern die Statistiken in eine .TFS-Datei. Sonst antworten Sie mit *Yes*.

Fehlermeldungen von Turbo Profiler

Turbo Profiler verwendet Fehlermeldungen, um Ihnen mitzuteilen, daß etwas Unerwartetes passiert ist. Manchmal kann der eingegebene Befehl nicht ausgeführt werden. In anderen Fällen werden Sie gewarnt, daß irgendetwas nicht ganz so funktioniert hat, wie Sie sich das gedacht haben.

Fehlermeldungen werden normalerweise auch von einem Alarmton begleitet. Sie können den Alarmton über das Konfigurationsprogramm TFINST dauerhaft ausschalten.

Already recording, do you want to abort?

[Aufnahme läuft bereits, wollen Sie abbrechen?]

Die Aufnahme eines Tastaturmakros ist bereits aktiviert. Sie können kein weiteres Makro aufnehmen, solange die aktuelle Aufnahme noch nicht abgeschlossen ist. Drücken Sie [Y, um die Aufnahme des Makros zu beenden, oder [N], um die Aufnahme fortzusetzen.

Ambiguous symbol *symbol name*

[Symbol *Symbolname* nicht eindeutig]

Sie haben eine Elementfunktion oder den Namen eines Datenelements eingegeben, der von Turbo Profiler nicht exakt zugeordnet werden kann, da es mehrerer Instanzen gibt.

Dies kann passieren, wenn eine Elementbezeichnung in zwei oder mehreren vererbten Klassen vorkommt. Verwenden Sie den Klassenbezugsoperator ::, um das gewünschte Element explizit anzugeben.

Bad or missing configuration file name

[Name für Konfigurationsdatei ist falsch oder fehlt völlig]

Sie haben in der Kommandozeilenoption **-c** den Namen einer Datei angegeben, die nicht existiert. Es werden die Standardwerte für die Einstellungen verwendet.

Bad interrupt number entered

[Eingabe einer unzulässigen Interrupt-Nummer]

Sie haben eine unzulässige Nummer für einen Interrupt eingegeben. Für Interrupts sind lediglich die Nummern 9 bis FF zulässig.

Bad module name *module name*

[Falscher Modulname *Modulname*]

Der eingegebene Modulname existiert nicht.

Can't execute DOS command processor

[DOS-Kommandoprozessor kann nicht aufgerufen werden]

Es war entweder nicht genug Speicher frei, um den DOS-Kommandoprozessor aufzurufen, oder der Kommandoprozessor wurde nicht gefunden (die Variable COMSPEC fehlt oder ist falsch). Stellen Sie sicher, daß die Variable COMSPEC den richtigen Wert enthält, damit der DOS-Kommandoprozessor zu finden ist.

Can't find *Dateiname.DLL*

[Kann *Dateiname.DLL* nicht finden]

Sie haben versucht, ein Programm zu laden, das eine oder mehrere DLL-Dateien benötigt, aber TPROFW kann diese nicht finden. Stellen Sie sicher, daß sich die ausführbare Datei und die DLLs, die von dieser Datei benötigt werden, in einem Verzeichnis befinden. Laden Sie dann das Programm erneut.

Can't swap user program to disk

[Anwendungsprogramm kann nicht auf Laufwerk ausgelagert werden]

Das untersuchte Programm kann nicht auf das Laufwerk ausgelagert werden. Es steht wahrscheinlich nicht genug Speicherplatz auf dem Laufwerk zur Verfügung, um das Programm auszulagern. Sie können keine Dateien editieren oder DOS-Befehle ausführen, solange nicht Platz geschaffen wird.

Error loading Program

[Fehler beim Laden des Programms]

Ihr Programm kann nicht geladen werden. Das Format der EXE-Datei stimmt nicht mit dem Betriebssystem überein.

Error printing statistics

[Fehler beim Drucken der Statistiken]

Fehler bei der Übertragung zum Drucker. Überprüfen Sie, ob der Drucker Online ist und ausreichend Papier eingelegt ist.

Edit program not specified

[Editor-Programm nicht angegeben]

Sie wollten den lokalen Befehl *Edit* im Fenster *Module* oder *Disk File* aufrufen, aber Sie können die Datei nicht editieren, da Turbo Profiler nicht weiß, wie der Editor aufgerufen wird.

Rufen Sie das Programm TFINST auf, um einen Editor anzugeben.

Error reading statistics file

[Fehler beim Lesen der Statistikdatei]

Während des Ladens der erfaßten Statistiken trat ein Fehler auf. Stellen Sie sicher, daß das Laufwerk betriebsbereit ist.

Error saving configuration

[Fehler beim Speichern der Konfiguration]

Ihre Konfiguration kann nicht gespeichert werden. Entweder steht nicht mehr genügend Speicherplatz auf dem Laufwerk zur Verfügung oder im Stammverzeichnis sind keine Verzeichniseinträge mehr frei.

Sie können mit *File / DOS Shell* in DOS verzweigen und dort dann eine oder mehrere Dateien löschen, um Platz für die Konfigurationsdatei zu schaffen.

Error swapping in user program, program reloaded

[Fehler beim Auslagern des Anwendungsprogramms, Programm erneut geladen]

Beim Wiedereinlesen Ihres ausgelagerten Programms trat ein Fehler auf. Das bedeutet normalerweise, daß die Auslagerungsdatei versehentlich zerstört wurde.

Sie müssen Ihr Programm mit dem Befehl *Run / Program Reset* erneut laden, bevor Sie die Analyse fortsetzen können.

Error writing statistics file

[Fehler beim Schreiben der Statistikdatei]

Während des Schreibens der Datei, in der die erfaßten Meßdaten abgespeichert werden sollten, kam es zu einem Fehler. Wahrscheinlich steht nicht mehr genug Speicherplatz zur Verfügung.

Stellen Sie sicher, daß das Laufwerk betriebsbereit ist und genug Speicherplatz auf der Diskette frei ist.

Exception *n*, error code *n*[Ausnahme *n*, Fehlernummer *n*]

Turbo Profiler hat entweder eine ungültige Speicherreferenz oder eine ungültige Anweisung in Ihrem Programm entdeckt. Sie müssen den Fehler beheben, bevor Sie die Analyse fortsetzen.

Help file TFHELP.TFH not found

[Hilfedatei TFHELP.TFH nicht gefunden]

Sie haben Hilfestellungen verlangt, aber die Datei mit den Hilfestellungen konnte nicht gefunden werden. Stellen Sie sicher, daß die Hilfe-Datei im selben Verzeichnis wie Turbo Profiler steht.

Invalid number entered

[Unzulässige Zahl eingegeben]

Sie haben entweder eine negative Zeilennummer eingegeben oder die eingegebene Zahl enthält ein Zeichen, das keine Ziffer ist. Als Zeilennummer darf nur eine positive ganze Zahl eingegeben werden.

Invalid statistics file

[Unzulässige Statistikdatei]

Die Datei, aus der die bereits erfaßten Statistiken geladen werden sollten, hat nicht das korrekte Format. Stellen Sie sicher, daß der Name einer Datei angegeben wird, die mit Hilfe des Befehls *Statistics / Save* erzeugt wurde.

Invalid switch

[Ungültige Option]

Beim Laden des Programms wurde eine ungültige Kommandozeilenoption gefunden.

Invalid window message number

[Ungültige Windows-Meldungsnummer]

Die von Ihnen angegebene Windows-Meldung ist kein gültiger Meldungsname oder keine gültige Meldungsnummer. Achten Sie darauf, daß der Meldungsname richtig geschrieben ist (Bei Windows-Meldungsnamen wird zwischen Groß- und Kleinschreibung unterschieden).

Maximum number of areas has been reached

[Maximale Anzahl an Bereichen erreicht]

Es kein Platz mehr vorhanden, um einen Bereich einzufügen. Verwenden Sie den Befehl *Options / Number of Areas*, um den Speicherbereich für markierte Bereiche zu vergrößern.

Maximum number of interrupts being monitored

[Maximale Anzahl an Interrupts erreicht, die überwacht werden können]

Sie können keine weiteren Interrupts mehr überwachen. Sie haben bereits soviele Interrupts wie möglich in Turbo Profiler zur Überwachung angegeben. Verwenden Sie den Befehl *Remove* des lokalen Menüs, um einen der ausgewählten Interrupts zu entfernen, bevor Sie einen weiteren Interrupt einfügen können.

Maximum number of windows messages has been reached

[Maximale Anzahl an Windows-Meldungen erreicht]

Turbo Profiler setzt per Vorgabe einen maximalen Wert für die zulässige Anzahl an Windows-Meldungen. Sie erhalten diese Fehlermeldung, wenn Sie eine höhere Anzahl an Meldungen protokollieren möchten, als im Textfeld *Max Windows Messages* (zu erreichen über *Statistics / Profiling Options*) vorgegeben wurde.

Wenn dieser Fehler auftritt, nimmt Turbo Profiler keine weiteren Meldungen mehr in die Liste auf.

Sehen Sie sich den Wert in *Max Windows Messages* an, überlegen Sie sich einen neuen, geben Sie diesen ein und ergänzen Sie dann die entsprechenden Meldungen im Eingabefenster *Window Procedure Messages*.

NMI interrupt

[NMI-Interrupt]

Das Programm, das Sie analysieren, hat einen NMI (Nicht maskierbaren Interrupt) ausgelöst.

No help for this context

[Keine Hilfestellung in diesem Zusammenhang]

Sie haben *[F1]* gedrückt, um Hilfestellungen zu erhalten, aber Turbo Profiler konnten keinen passenden Hilfe-Bildschirm finden.

No file name was given

[Kein Dateiname angegeben]

Sie wollten eine Ausgabe in einer Datei erzeugen, haben aber keinen Dateinamen angegeben. Sie müssen entweder einen Dateinamen angeben oder zu einer anderen Ausgabestelle wechseln, bevor Sie den Dialog verlassen können.

No modules with statistics

[Keine Module mit Statistiken]

Es gibt keine Module, für die Meßdaten erfaßt wurden, also kann auch nichts ausgedruckt werden.

No network present

[Kein Netzwerk vorhanden]

Sie müssen NETBIOS (Version 3.0 oder höher) laden, bevor Sie TPROF oder TFREMOTE mit der Option **-rn** starten.

No previous search expression

[Kein vorheriger Suchausdruck vorhanden]

Sie haben den Befehl *Next* in einem Textausschnitt angegeben, ohne daß zuvor ein *Search*-Befehl ausgeführt wurde. Geben Sie zunächst den Befehl *Search* mit dem gesuchten String ein. Anschließend können Sie die Suche mit *Next* wiederholen.

No program loaded

[Kein Programm geladen]

Sie wollten einen Befehl eingeben, bei dem ein Programm geladen sein muß. Es gibt viele Befehle, die nur dann eingegeben werden können, wenn ein Programm geladen ist, zum Beispiel die Befehle im *Run*-Menü. Verwenden Sie den Befehl *File/Open* um ein Programm zu laden, bevor Sie diese Befehle einsetzen.

No source file for module *Modulname*[Keine Quelldatei für das Modul *Modulname*]

Die Quelldatei kann für das angegebene Modul nicht gefunden werden. Die Quelldatei wird zuerst im aktuellen Verzeichnis und dann in den Verzeichnissen gesucht, die in der Konfigurationsdatei angegeben wurden,

anschließend in den Verzeichnissen, die nach der Kommandozeilenoption *-sd* stehen.

Not a code address

[Keine Code-Adresse]

Sie haben eine Adresse eingegeben, die in Ihrem Programm nicht vorkommt. Bereiche können nur an Programmadressen markiert werden.

Not available when in coverage mode

[Im Coverage-Modus nicht verfügbar]

Diese Fehlermeldung tritt auf, wenn Sie versuchen, eine Funktion zu verwenden, die im Coverage-Modus nicht funktioniert. Wahrscheinlich haben Sie versucht, eine Bereichsmarkierung zu setzen oder zu entfernen. Da Turbo Profiler die Markierungen für den Coverage-Modus selbst setzt, können Sie diese nicht per Hand setzen oder entfernen.

Not enough memory for selected operation

[Nicht genug Speicher für die ausgewählte Operation]

Sie haben einen Befehl eingegeben, bei dem ein Fenster angelegt werden muß. Für diese Operation ist aber nicht genug Speicher frei, so daß Sie zuerst ein anderes Fenster schließen oder verkleinern müssen, bevor Sie diesen Befehl wiederholen können.

Not enough memory to load program

[Nicht genug Speicher zum Laden des Programms]

Die Symboltabelle Ihres Programms konnte korrekt in den Speicher geladen werden, aber es ist nicht mehr genug Speicher frei, um Ihr Programm zu laden. Sie können zwei Systeme miteinander verbinden und Turbo Profiler auf dem einen System und das Programm auf dem anderen System starten. Weitere Informationen dazu finden Sie in Anhang C.

Not enough memory to load symbol table

[Nicht genug Speicher zum Laden der Symboltabelle]

Es ist nicht genug Platz vorhanden, um die Symboltabelle Ihres Programms in den Speicher zu laden. Die Symboltabelle enthält Informationen, die Turbo Profiler für die Anzeige Ihres Quellcodes und der Programmvariablen benötigt. Wenn Sie speicherresidente Hilfsprogramme geladen haben, sollten Sie diese entfernen und Turbo Profiler erneut starten. Sie können auch die Symboltabelle verkleinern, indem Sie den Compiler anweisen, die Symbolinformationen nur für die Module zu erzeugen, die Sie untersuchen wollen.

Wenn diese Meldung ausgegeben wurde, ist ihr Programm noch nicht geladen. Das heißt, Sie sollten genügend Speicher zur Verfügung stellen, um sowohl die Symboltabelle als auch das Programm laden zu können.

Out of heap space

[Keinen Heap-Speicher frei]

Turbo Profiler hat keinen Speicher mehr frei, als er versuchte, Ihre Informationen, die Sie benötigen, zu sammeln. Sie müssen die Datenmenge, die Sie in einem Durchlauf sammeln möchten, verringern.

Overlay not loaded

[Overlay nicht geladen]

Sie wollten Code in einem Overlay untersuchen, das nicht in den Speicher geladen wurde. Sie können nur Programmcode analysieren, der im Speicher liegt.

Sie können sich aber trotzdem den Quellcode des gewünschten Moduls im *Module*-Fenster anzeigen lassen.

Overwrite existing macro on selected key

[Vorhandenes Makro für den ausgewählten Tasten-Code überschreiben]

Sie haben versucht, ein Makro einer bestimmten Tastenkombination zuzuordnen, der jedoch bereits ein Makro zugewiesen ist. Wenn Sie das bestehende Makro überschreiben wollen, drücken Sie [Y], sonst drücken Sie [N], um den Befehl zu widerrufen.

Overwrite file name?

[Dateinamen überschreiben?]

Sie haben den Namen einer Datei angegeben, die bereits existiert. Mit [Y] können Sie die Datei überschreiben und den alten Inhalt ersetzen. Um den Befehl abubrechen, geben Sie [N] ein. Die alte Datei bleibt auf diese Weise unverändert erhalten.

Path not found

[Pfad nicht gefunden]

Sie haben eine Kombination aus Laufwerk und Verzeichnis angegeben, die nicht existiert. Überprüfen Sie, ob Sie die Bezeichnung des gewünschten Laufwerks und des Verzeichnisses richtig geschrieben haben.

Das aktuelle Laufwerk und Verzeichnis ist nicht verändert worden.

Premature end of string in Symbolname[Vorzeitiges Ende des Strings in *Symbolname*]

Der eingegebene Symbolname ist nicht vollständig. Wenn Sie einen Modulnamen angeben, muß dahinter eine Zeilennummer oder ein lokaler Symbolname stehen.

Press key to assign macro to

[Taste drücken, der ein Makro zugewiesen werden soll]

Drücken Sie die Tastenkombination, der Sie ein Makro zuweisen wollen. Anschließend geben Sie die Befehlsfolge ein, die der Makro-Taste zugewiesen werden soll. Die Befehlsfolge wird genauso ausgeführt, wie Sie von Ihnen eingegeben wurde. Um die Aufnahme des Makros zu beenden,

drücken Sie die Tastenkombination, der das Makro zugewiesen wird, ein zweites Mal. Dieses Makro wird zusammen mit den anderen Makros in einer Datei gespeichert.

Procedure stack overflow

[Stapelüberlauf in der Prozedur]

Ihr Programm hat zu viele verschachtelte Prozedur- oder Funktionsaufrufe. Sie müssen einige Bereiche entfernen, die in den untersten Aufrufpfaden der Routinen liegen. Verwenden Sie das Fenster *Callers*, um diese Stellen zu finden.

Program does not have overlay

[Programm hat keine Overlays]

Das von Ihnen gerade untersuchte Programm hat keine Overlays, so daß das *Overlays*-Fenster nicht geöffnet werden kann.

Program has invalid symbol table

[Programm hat eine unzulässige Symboltabelle]

Das Programm, das Sie laden wollen, hat eine Symboltabelle mit einem nicht zulässigen Format. Erzeugen Sie eine neue ».EXE«-Datei und laden Sie das Programm neu.

Program has no symbol table

[Programm hat keine Symboltabelle]

Das Programm, das Sie analysieren wollen, wurde korrekt geladen, aber es enthält keine Symbolinformationen. Linken sie das Programm erneut, damit es eine Symboltabelle erhält.

Program linked with wrong linker version

[Programm wurde mit falscher Linker-Version gelinkt]

Das Programm, das Sie laden wollen, wurde mit einer alten Version des Linkers gelinkt. Sie müssen die neueste Version des Linkers verwenden, um Programme analysieren zu können.

Program not found

[Programm nicht gefunden]

Das Programm, das Sie laden wollen, existiert nicht. Überprüfen Sie, ob der Name im Befehl *File / Open* richtig eingegeben und eine andere Datei-erweiterung als .EXE angegeben wurde.

Program out of date or missing on remote, send over link?

[Programm im Remote-System nicht aktuell, erneut senden?]

Sie haben ein Programm für die Analyse auf dem Remote-System ausgewählt, aber entweder existiert dieses Programm dort nicht oder die Datei auf dem lokalen System ist aktueller als diejenige auf dem Remote-System.

Wenn Sie [Y] drücken, wird das Programm über die Verbindung übertragen. Drücken Sie aber [N], wird das Programm nicht geschickt und der Befehl *File/ Open* wird abgebrochen.

Normalerweise werden Sie mit [Y] antworten. Wenn Sie die Verbindung (mit der Kommandozeilenoption **-rs1** auf die langsamste Übertragungsgeschwindigkeit eingestellt haben, können Sie den Befehl mit [N] abbrechen und die Datei mit Hilfe einer Diskette auf das Remote-System übertragen.

Reload program so arguments take effect?

[Programm erneut laden, damit die Argumente übernommen werden?]

Die meisten Programme müssen erneut geladen werden, wenn Ihre Kommandozeilenargumente geändert werden. Drücken Sie auf diese Frage im Zweifelsfall immer [Y].

Reload program so new area count takes effect?

[Programm neu laden, um die neuen Bereiche zu berücksichtigen?]

Damit Turbo Profiler den Speicherbereich für die Statistik-Daten reservieren kann, muß Ihr Programm aus dem Speicher entfernt werden und von Anfang an noch einmal ausgeführt werden.

Drücken Sie [Y], damit dies geschieht. Falls Sie [N] drücken, müssen Sie auf die nächste Gelegenheit warten, um das Programm manuell nachzuladen, damit die neuen Bereichsgrößen zum Einsatz kommen.

Run out of space for keystroke macros

[Nicht genug Platz für die Tastatur-Makros]

Es ist nicht genug Speicherplatz für die Tastatur-Makros vorhanden.

Search expression not found

[Gesuchten Ausdruck nicht gefunden]

Der angegebene Textstring oder die Bytefolge ist in der Datei nicht vorhanden. Da die Suche von der aktuellen Cursor-Position nach vorne erfolgt, sollten Sie noch einmal (mit dem Kurzcode [Strg][Bild oben]) an den Anfang der Datei zurückgehen und die Suche wiederholen.

Stopped by area

[Angehalten im Bereich]

Turbo Profiler hat einen Bereich gefunden, dessen Operation Sie auf *Stop* gesetzt haben. Sie können die Analyse mit Hilfe des Befehls *Run / Run* fortsetzen.

Symbol not a routine name

[Symbol ist keine Routinenbezeichnung]

Die angegebene Symbolbezeichnung ist kein zulässiger Name einer Routine.

Symbol not found

[Symbol nicht gefunden]

Sie haben einen Ausdruck mit einer unzulässigen Symbolbezeichnung angegeben. Als Symbol zulässig ist:

Ein Name eines globalen Symbols

Eine Modul-Bezeichnung, hinter der der Name eines lokalen Symbols steht

Eine Modul-Bezeichnung auf die eine dezimale Zeilennummer folgt.

Syntax error in Symbol *Symbolname*

[Syntaxfehler in Symbol *Symbolname*]

Sie haben eine unzulässige Bezeichnung für ein Symbol eingegeben. Als Symbol zulässig ist:

Ein Name eines globalen Symbols;

Eine Modul-Bezeichnung, hinter der der Name eines lokalen Symbols steht;

Eine Modul-Bezeichnung, auf die eine dezimale Zeilennummer folgt.

Too many areas for a Windows programm

[Zu viele Bereiche für ein Windows-Programm]

Sie versuchen, ein Programm zu analysieren, das mehr als die 551, von Windows unterstützten Bereiche hat. Einige der von Ihnen eingestellten Bereiche werden nicht analysiert. Sie müssen die Anzahl der Bereiche verringern, um zu kontrollieren, welche Bereiche nicht eingeschlossen werden sollen.

Too many files match wildcard mask

[Zuviele Dateinamen stimmen mit Jokerzeichenangabe überein]

Sie haben eine Jokerzeichen-Maske angegeben, mit der mehr als 100 Dateien ausgewählt werden. In der Liste können aber nur die ersten hundert Dateien dargestellt werden.

Unknown control point

[Unbekannter Kontrollpunkt]

Turbo Profiler hat eine *INT3*-Anweisung in Ihrem Programm entdeckt, die nicht zugelassen ist. Turbo Profiler verwendet *INT3*-Anweisungen, um Kontrollpunkte festzulegen. Wenn Sie selbst diese Anweisung in Ihr Programm einfügen, entstehen Konfliktsituationen.. Es ist auch möglich, daß Turbo Profiler einen Kontrollpunkt eingefügt und ihn dann verloren hat.

Wenn Sie eine *INT3*-Anweisung in Ihr Programm eingefügt haben, müssen Sie diese entfernen, damit Sie mit Turbo Profiler dieses Programm weiter analysieren können.

Wenn Sie keine *INT3*-Anweisung eingefügt haben, so liegt dies an Turbo Profiler. Entfernen Sie den Bereich, in dem sich die *INT3*-Anweisung befindet, und Sie können die Analyse fortsetzen.

Unable to determine procedure type

[Kann Prozedurtyp nicht bestimmen]

Turbo Profiler kann nicht bestimmen, ob der momentane Bereich eine Prozedur vom Typ *far* oder *near* ist. Sie müssen den Bereich aus der Bereichsliste entfernen, damit Turbo Profiler fortfahren kann.

Value must be between *nn* and *nn*

[Wert muß zwischen *nn* und *nn* liegen]

Sie haben einen ungültigen Wert für eine Editor-Einstellung (etwa für die Tabulatoreinstellung) oder für eine Druckereinstellung (z. B. für die Anzahl der Zeilen pro Seite) angegeben. Die Fehlermeldung gibt auch den zulässigen Bereich der einzugebenden Werte aus.

Video monitor switched while flipping pages

[Video-Monitor während Seitenwechsel umgeschaltet]

Sie haben Turbo Profiler mit einer Video-Einstellung gestartet, in der Bildschirmdarstellungen nicht gespeichert werden. Das Programm, das Sie untersuchen, hat in einen Grafik-Modus umgeschaltet.

Turbo Profiler hat die Bildschirmdarstellung in den Text-Modus zurückgeschaltet, so daß die Ausgaben des Programms verloren sind.

Um dies zu vermeiden, sollten Sie Turbo Profiler mit Bildschirmumschaltung (Kommandozeilenoption *-ds*) aufrufen.

Waiting for remote driver. Press Esc to stop waiting

[Warte auf den Remote-Treiber. Drücken Sie *[Esc]*, um abzubrechen]

Sie haben eine Remote-Analyse gestartet, und Turbo Profiler wartet auf eine Antwort des Remote-Treibers. Drücken Sie *[Esc]*, um die Remote-Sitzung abzubrechen.

Index

A

- About (System) 41
- Accumulation
 - Statistics 81
- Add Areas
 - All Areas 48
 - All Routines 66
 - Areas-Fenster 65
 - Current Routine 48
 - Every Line in Module 48
 - Lines in Routine 48
 - Module 66
 - Module-Fenster 48
 - Modules with Source 48
 - Routines in Module 48
 - This 48
- Akkumulation 81
 - ausschalten 82
- aktive Analyse 80
- Algorithmen
 - testen 104
- All Areas
 - hinzufügen 48
- Analyse
 - Beispiel 17
- Analysen
 - aktive 80, 106
 - Anzeige des Modus 44
 - Details 100
 - laden 78, 103
 - Modus 106
 - Modus einstellen 80
 - Optionen 80
 - passive 80, 107
 - schnellere 108
 - selektieren 54
 - speichern 78, 103
 - Strategien 95
 - vorbereiten 96
 - Zeittaktfrequenz 81
- Anzahl
 - anzeigen 54
 - der Programmläufe 80

- Area Operation 49
- Areas
 - All Routines 51
 - Callers 50
 - Routines-Fenster 68
 - This Module 51
 - This Routine 50, 51
- Areas-Fenster 64
 - Add Areas 65
 - Inspect 66
 - Options 66
 - Remove Areas 66
 - Sort 67
- ASCII-Zeichensatz 141
- Aufruf
 - Optionen 132
 - TPROF 129
- Aufruf-Stack 126
- Aufrufhierarchie 105, 126
- Aufrufinformationen 105, 125
 - Callers-Fenster 56
 - direkte Aufrufe 67
 - einschalten 78
 - Module-Fenster 50
 - sortieren 59
- Ausführungen
 - Anzahl anzeigen 54
- Ausführungszeit 105
 - anteilige 52
 - anzeigen 53
 - gesamte 52
 - und Anzahl anzeigen 54
- Ausschnitt
 - wechseln 39, 92

B

- Baudrate 133
- Beispiel 17
- Benutzerbildschirm 93
- Benutzeroberfläche 35
 - Bestandteile 35
- Bereiche 99
 - aktuelle Routine markieren 48

- aktuelle Zeile entfernen 48
- alle Routinen des Moduls hinzufügen 48
- alle Routinen markieren 48
- alle Routinen mit Quellcode hinzufügen 48
- alle Zeilen der Routine hinzufügen 48
- alle Zeilen des Moduls hinzufügen 48
- Effizienz messen 28
- einfügen 48
- entfernen 66
- festlegen 98
- hinzufügen 65
- Informationen speichern 96
- markieren 48
- Markierungen 64
- Operation 66
- Operationen 49, 99
- Optionen 66
- Routinen 68
- sortieren 67
- Bereichsdatei 96
- Bereichsinformationen 65, 84
- Bereichsmarkierung 99
- Bereichsmarkierungen 48
 - alle Routinen mit Quellcode hinzufügen 48
- Bereichsoperationen 49
 - Operation 49
 - Timing 49, 50
- Betriebssystem-Shell 44
- Bildlaufleiste 37
- Bildschirm
 - 25 vs. 43/50 Zeilen 89
 - 43/50-Zeilenmodus 134
 - Anordnung speichern 91
 - Auslagerung 89
 - EGA-Palette 134
 - Einstellung 143
 - Farb- 132
 - Farben 137
 - Grafikspeicher 134
 - Monochrome- 132
 - neu aufbauen 40
 - Optionen 89, 134
 - Tabulator 90
 - zweiter 132
- Bildschirmanordnung
 - Standardeinstellung 40
- Bildschirmseite
 - zweite 132
- BIOS-Adressen 70
- Borland C++

Debug-Informationen 18, 98

C

- Callers 126
 - Module-Fenster 50
 - Routines-Fenster 68
 - Statistics 78
- Callers-Fenster 56, 112
 - Inspect 59
- Coverage-Analyse 73
- Clock Speed 81
- Close
 - Window 93
- Collection 44
- COM1 133
- COM2 133
- Combined 50
- Control-Tastenkürzel 141
- Coverage-Analyse
 - Module 76
 - Position 76
- Coverage-Analyse
 - Add All Modules 74
 - Add Module 74
 - Delete This Item 75
 - Display 75
 - Group 75, 76
 - Remove All Modules 74
 - Remove Module 75
 - Sort 75
- Coverage-Analysis 80
- Coverage-Fenster 113
- Create
 - Macros 87
- Current Routine
 - hinzufügen 48

D

- Darstellungsmodus 132
- Datei laden
 - Text File-Fenster 73
- Dateien
 - laden 41
 - öffnen 41
- Dateizugriffe
 - Datenüberlauf 64
 - detailliert 64

- erfassen 63, 79
- Datum 44
- Debug-Informationen
 - Borland C++ 18, 98
 - Turbo Assembler 18, 98
- Delete All
 - Macros 88
 - Statistics 84
- Destination File 87
- Dialogfenster 39
 - Area Operation 49
 - Directories (TFINST) 140
 - Display options 53, 89
 - Meldung 173
 - Miscellaneous Options (TFINST) 141
 - Pick a Caller 59
 - Pick a Module 68
 - Printing options 87
 - Profiling Options 80
 - Program load 41
 - Restore 91
 - Save Configuration 90
 - User Input & Prompting (TFINST) 140
- Directories (TFINST) 140
- Disassembly-Fenster 69
 - Follow 70
 - Goto 70
 - Origin 70
 - Previous 71
 - View Source 71
- Display
 - Both 54
 - Count (Overlays) 60
 - Counts 54
 - Execution Profile 53
 - History (Overlays) 61
 - Longest 54
 - Modules 54
 - Overlays 60
 - Per Call 54
 - Time 53
- Display Options 53, 89
- DLLs
 - Analyse 169
- DOS
 - Adressen 70
 - aufrufen 44
 - Größe für Auslagerung 142
 - Speicherbelegung 44
- DOS Shell 44

- DOS-Interrupts 61
- Druck-Datei 87
- Druck-Optionen 87
- Drucken
 - Breite einstellen 87
 - Datei 87
 - Höhe einstellen 87
 - Module 86
 - Optionen 87
 - Statistiken 86
- Drucker
 - Installation 141

E

- Eckfeld zur Änderung der Größe 37
- Editor
 - aufrufen 52
 - installieren 140
- Effizienz
 - noch mehr 33
- EGA-Palette 134
- Eingabeanforderungen 173
- Eingabeaufzeichnungslisten
 - Länge 141
- Erfassung
 - Anzeige 44
 - Beispiel 20
 - Einstellung 78
 - Frequenz 81
 - Optionen 80
 - starten 101
- Every Line in Module 48
- Execution Profile 52, 112
 - Art der Darstellung 53
 - Display 53
 - Filter 54
 - Module 55
 - Remove 56

F

- Farbbildschirm 132, 143
- Farbeinstellung
 - Hintergrund 137
- Fehlermeldungen 177
 - TFREMOTE 158, 159, 160
 - WREMOTE 161, 162
- Fenster 37

- aktives 37
- Areas 64
- Aufrufinformationen 56
- Ausschnitt wechseln 39, 92
- Bedienung 38
- Bereiche 64
- Callers 56, 112
- Coverage 113
- Dateizugriffe 63
- Disassembly 69
- einstellen 92
- Einstellung 136
- Einstellung speichern 27
- Execution Profile 52, 112
- Files 63, 112
- Größe ändern 38
- in Symbol verwandeln 93
- Inhalt 38
- Interrupts 61, 112
- Liste der Fenster 93
- Module 46
- nächstes 92
- Overlays 60, 112
- Reports 111
- Routines 67
- Text File 72
- Umwandlung in Symbol 39
- vergrößern 92
- vergrößern/verkleinern 39
- verkleinern 92
- verschieben 38, 92
- Verwaltung 38
- View-Menü 45
- wiederherstellen 93
- zoomen 92
- Fensterelemente
 - Bildlaufleiste 37
 - Eckfeld zur Änderung der Größe 37
 - Fensternummer 37
 - Schließfeld 37
 - Symbolfeld 37
 - Titel 37
 - Zoom-Feld 37
- Fensternummer 37
- File
 - DOS Shell 44
 - Get Info 43
 - Menü 41
 - Module-Fenster 52
 - Open 41

- Quit 44
- Text File-Fenster 73
- Files
 - Statistics 79
- Files-Fenster 63, 112
 - Collection 63
 - Detail 64
- Filter
 - All 54
 - Current 55
 - Execution Profile 54
 - für Meßergebnisse 113
 - Module 54
- Follow
 - Disassembly-Fenster 70
- Fragen 15

G

- Genauigkeit
 - statistische 109
- Get Info 43
- Goto
 - Disassembly-Fenster 70
 - Module-Fenster 48
 - Text File-Fenster 72
- Grafikprogramme 134
- Großschreibung 133, 142

H

- Hardware-Voraussetzungen
 - Turbo Profiler 13
- Hauptspeicher 44
- Help 93
 - Index 94
 - on Help 94
 - Previous Topic 94
- Hilfestellungen 93, 132
 - Index 94
 - zur Hilfe 94
- Hotkeys 37

I

- Iconize/Rstore
 - Window 93
- Index
 - Help 94

- Informationen
 - allgemeine 41
 - System- 43
- Inspect
 - Areas-Fenster 66
 - Caller-Fenster 59
- Installation 135
- Interrupt-Taste 77
- Interrupts 61
 - Anzahl der Aufrufe 62
 - aus Liste auswählen 62
 - aus Liste entfernen 62
 - chronologische Liste 62
 - Darstellung 62
 - erfassen 79
 - Erfassung einschalten 62
 - Files 79
 - in Liste einfügen 62
 - Liste vollständig löschen 62
 - Statistics 79
 - Unterrouinen 62
 - Zeitdauer erfassen 62
- Interrupts-Fenster 61, 112

K

- Kleinschreibung 133, 142
- Kommandozeilenoptionen 129
 - ? 132
 - b 130
 - c 131
 - d 132
 - do 132
 - dp 132
 - ds 132
 - h 132
 - jn 132
 - p 132
 - r 133
 - rnl 133
 - rp 133
 - rs 133
 - s 133
 - sc 133
 - sd 133
 - vn 134
 - vp 134
 - vs 134
- Darstellungsmodus 132

- Konfigurationsdatei 131
- Stapelverarbeitungsbetrieb 130
- TPROFW 165
- Übersicht 129
- und TFINST 145
- Konfiguration
 - laden 91
 - speichern 90
- Konfigurationsdatei 91, 131
 - laden 144
 - speichern 144

L

- Label
 - auswählen 48
- LCD-Bildschirm 144
- Line
 - Module-Fenster 47
- Lines in Routine 48
- Lokal Menüs
 - Auswahl 45
- Longest
 - Display (Statistiken) 54

M

- Macros
 - Create 87
 - Delete All 88
 - Options 87
 - Remove 88
 - Stop recording 88
- Makros 87
 - abschließen 88
 - alle löschen 88
 - erstellen 87
 - löschen 88
 - speichern 88, 91
- Markierungen 84
 - aktuelle Routine hinzufügen 48
 - aktuelle Zeile entfernen 48
 - aktueller Zeile löschen 56
 - alle Bereiche des Moduls hinzufügen 48
 - alle Routinen hinzufügen 48
 - alle Zeilen des Moduls hinzufügen 48
 - Aufrufinformationen 67
 - durchführen 98
 - entfernen 66

- hinzufügen 65
- Informationen speichern 96
- sortieren 67
- Zeilen der Routine hinzufügen 48
- Maus
 - Hotkeys 37
 - Lokale Menüs auswählen 45
- Mausunterstützung 132, 141
- Memory 44
- Menüleiste 35
- Menüoptionen
 - Tastatur 36
- Menüs 40
 - File 41
 - globale 35
 - Help 93
 - Options 87
 - Print 85
 - Referenz 40
 - Run 77
 - Statistics 78
 - System 40
 - View 45
 - Window 92
- Meßergebnisse 52
 - Art der Darstellung 53
 - Einstellungen 78
 - entfernen 55
 - filtern 113
 - laden 103
 - nach Adressen sortieren 54
 - nach Häufigkeit sortieren 54
 - nach Namen sortieren 54
 - selektieren 54
 - speichern 103
- Miscellaneous Options (TFINST) 141
- Modify
 - TPROF.EXE 144
- Module
 - andere Quelltexte eines M. 51
 - ausdrucken 23
 - Display 54
 - drucken 86
 - Filter 54
 - laden 52
 - Module-Fenster (lok. Befehl) 51
 - Print 86
 - Wechsel von Statistiken 55
- Module-Fenster 46
 - Add Areas 48

- Callers 50
- File 52
- Goto 48
- Line 47
- Module (lok. Befehl) 51
- Next 48
- Operation 49
- Remove Areas 49
- Search 47
- Modules with Source
 - hinzufügen 48
- Monochrome-Bildschirm 132, 144

N

- Neustart 77
- Next
 - Module-Fenster 48
 - Text File-Fenster 73
 - Window 92
- Next Pane
 - Window 92
- NMI 142

O

- objektorientierte Programmierung 111
- Open
 - File 41
- Operation 49
 - Normal 49
 - Windows 50
- Operationen 99
 - Operation 49
 - Timing 49, 50
- Operationen (Bereiche) 49
- Optimierung 12, 116
- Optionen 87
 - Bildschirm 89
 - drucken 87
 - Einstellung 20
 - laden 91
 - speichern 90
- Options 87
 - Areas-Fenster 66
 - Display Options 89
 - Macros 87
 - Path for Source 90
 - Print 87

- Restore 91
- Save 90
- Origin
 - Disassembly-Fenster 70
- Overlays 60, 110
 - Ablaufdarstellung 61
 - Anzahl der Aufrufe 60
 - erfassen 79
 - Files 79
 - Inspect 61
 - Statistics 79
 - Statistiken 60
 - untersuchen 61
- Overlays-Fenster 60, 112

P

- passive Analyse 80
- Path for Source 90
- Pick a Caller 59
- Pick a Module 68
- Position 56
- Previous
 - Disassembly-Fenster 71
- Previous Topic
 - Help 94
- PRIME0 19
- PRIME1 29
- PRIME2 31
- PRIME3 31
- PRIME4 32
- PRIME6 33
- Primzahlen 18
- Print 85
 - Module 86
 - Options 87
 - Statistics 86
- Printing options 87
- Probleme 15
- Profile 53
 - Normal Areas 53
 - Window Procs 53
- Profiler
 - beenden 44
 - Listing 23
 - Speicherplatzbelegung 128
 - Status 36
- Profiling Options 80
 - Clock Speed 81
- Profile Mode 80
- Run Count 80
- Program load 41
- Programm
 - Analyse 103
 - ändern 30
 - anhalten durch Operation 49
 - anpassen 97
 - Anzahl der Aufrufe 52
 - Argumente 77
 - beenden 44
 - Compilierung 98
 - laden 46
 - mehrmals analysieren 81
 - modularisieren 29
 - Name 43
 - neu starten 77
 - objektorientierte 111
 - Parameter 77
 - Position 70
 - PRIME2 laden 31
 - starten 77
 - Status 43
 - Struktur untersuchen 105
 - überarbeiten 114
 - Umstrukturierung 59
- Programmanalyse 52
- Programmaufruf 129
- Programmende 44
- Protokolle 132

Q

- Quelltext
 - zusammen mit Statistiken 26
- Quelltexte 133
 - andere Q. laden 51
 - im Module-Fenster 46
 - Suchverzeichnisse 46
- Quit 44

R

- Remote Type 143
- Remote-Analyse 147
 - DOS-Anwendungen 149
 - Windows-Anwendungen 147, 153
 - Lokale Netzwerke 133
 - Optionen 149

- Windows-Programme 155
- Remote-Analyse:Hardware- u. Software 148
- Remote-Analyse:Installation von TFREMOTE 149
- Remote-Analysen 143
 - Baudrate 143
 - Optionen 132
 - Schnittstelle 133
 - Standardeinstellung 133
 - Übertragungsgeschwindigkeit 133
- Remove
 - Execution Profile 56
 - Macros 88
- Remove Areas
 - All Areas 66
 - Areas-Fenster 66
 - Module 66
 - Module-Fenster 49
 - This Area 66
- Repaint Desktop (System) 40
- Report-Fenster 111
- Reporte 25
- Resonanzen 109
- Ressourcen 105
- Restore 91
 - Options 91
 - Statistics 85
- Restore Standard (System) 40
- Routinen
 - alle im Modul markieren 48
 - alle mit Quelltext markieren 48
 - Aufrufinformationen 68
 - Aufwand für Aufrufe 121
 - auswählen 48
 - Bereiche 68
 - Fenster 67
 - globale 68
 - Länge der Bezeichner 90
 - lokale 68
 - markieren 48
 - Optimierung 117
 - untersuchen 59
- Routines in Module
 - hinzufügen 48
- Routines-Fenster 67
 - Areas 68
 - Callers 68
 - Local Module 68
- Run 77
 - Arguments 77

- Program reset 77
- Run (Befehl) 77
- Run Count 80

S

- Save
 - Options 90
 - Statistics 84
- Schleifen 116
 - Aufwand 121
- Schließfeld 37, 38
- Schnittstelle 133
- Schwarzweißbildschirm 143
- Search
 - Module-Fenster 47
 - Text File-Fenster 72, 73
- Shell
 - aktivieren 44
- Size/Move
 - Window 92
- Software-Voraussetzungen
 - Turbo Profiler 13
- Sort
 - Address 54
 - Areas-Fenster 67
 - Called 59
 - Callers 59
 - Frequency 54
 - Frequency (Callers) 59
 - Meßergebnisse 54
 - Name 54
- Sortieren
 - Aufrufinformationen 59
 - Bereiche 67
 - Meßergebnisse 54
- Sortierreihenfolge 142
- Speicherplatzbelegung 128
 - Hauptspeicher 44
- Stack
 - All Callers 51
 - Callers 51
 - Immediate Callers 51
 - None 51
- Stapelverarbeitungsbetrieb 130
- Statistics 78
 - Accumulation 81
 - Callers 78, 126
 - Delete All 84

- Files 79
- Print 86
- Profiling Options 80
- Restore 85
- Save 84
- Statistiken 52, 78
 - akkumulieren 81
 - ausdrucken 23
 - auswerten 26
 - darstellen 21
 - drucken 86
 - Erfassung ein-/ausschalten 49
 - laden 25, 103
 - löschen 84
 - Overlays 60
 - selektieren 54
 - speichern 25, 84, 103
- Status 36
- Statuszeile 39
- Stichproben 127
- Stop recording
 - Macros 88
- Strategien 95
- String suchen 72, 73
- Stringsuche 47
 - wiederholen 48
- Suche
 - Module-Fenster 47, 48
- Suche wiederholen
 - Text File-Fenster 73
- Suchen
 - Text File-Fenster 72, 73
- Suchverzeichnisse 46, 90, 133, 140
- Symbol
 - in Fenster verwandeln 93
- Symbolfeld 37
- Symboltabellen 133
- System
 - About 41
 - Menü 40
 - Repaint Desktop 40
 - Restore Standard 40

T

- Tabulator 90
- Taktfrequenz 81
- Tastatur
 - Hotkeys 37

- Tastenkürzel 37
 - Control- 141
- TDDEBUG.386
 - Installation 164
- TDRF 149
- Testen 104
- Text File-Fenster 72
 - File 73
 - Goto 72
 - Next 73
 - Search 72, 73
- .TFA-Dateien 96
- TFCONFIG.TF 91
- TFINST 135
 - und Kommandozeilenoptionen 145
- TFREMOTE 149
 - Aufruf 150
 - Fehlermeldungen 158, 159, 160
 - Kommandozeilenoptionen 149
- .TFS-Dateien 78, 85, 103
- This 48
- Timing 49, 50
 - Separate 50
- Titel eines Fensters 37
- TPROF
 - Aufruf 129
- TPROF.EXE
 - Modify 144
- TPROFW 163
- Turbo Assembler
 - Debug-Informationen 18, 98
- Turbo directory 140
- Turbo Profiler
 - Aufruf 129
 - Fehlermeldungen 177
 - Funktionsweise 119
 - Installation 135
 - Kommandozeilenoptionen 129
 - Speicherplatzbelegung 128

U

- Uhrzeit 44
- Undo Close
 - Window 93
- Unterbrechungstaste 141
- User Input & Prompting (TFINST) 140
- User screen
 - Window 93

V

- Verifikation 104
- View
 - Textfile 72
- View Source
 - Disassembly-Fenster 71

W

- Window 92
 - Close 93
 - Iconize/Restore 93
 - Next 92
 - Next Pane 92
 - Size/Move 92
 - Undo Close 93
 - User screen 93
 - Zoom 92
- Windows-Anwendungen
 - Remote-Analyse 153
- WREMOTE
 - Fehlermeldungen 161, 162
 - Kommandozeilenoptionen 155

Z

- Zählen 127
- Zeilen
 - aktuelle Z. entfernen 48
 - Anzahl der Zeilen 134
 - in Modul markieren 48
 - in Routine markieren 48
- Zeilennummer
 - springen zu 47, 72
- Zeit/Anzahl anzeigen 54
- Zeitdaten
 - Darstellung 102
- Zeitdauer
 - anzeigen 53
 - längste pro Aufruf anzeigen 54
 - pro Aufruf anzeigen 54
 - und Anzahl anzeigen 54
- Zeittakte 81
 - anzeigen 52
- Zeittaktfrequenz 81
- Zoom 39
 - Window 92
- Zoom-Feld 37
- Zoomen 92

Borland

Copyright © 1994, Borland International, Inc. Alle Rechte vorbehalten. Alle Produktnamen von Borland sind eingetragene Warenzeichen der Borland International Inc.

Deutsche Ausgabe © 1994 Borland GmbH, Monzastraße 4c, D-63225 Langen, Telefon 06103/979-0, Fax 06103/979-290

Zweigniederlassung Schweiz, Solothurnstraße 31, CH-3303 Jestorf

Hauptsitz: 100 Borland Way, Scotts Valley, CA 95066-3249, +1 (408) 431-1000

Niederlassungen in: Australien, Belgien, Brasilien, Chile, Deutschland, Dänemark, Frankreich, Großbritannien, Hongkong, Italien, Japan, Kanada, Korea, Lateinamerika, Malaysia, Niederlande, Neuseeland, Schweden, Singapur, Spanien und Taiwan.

• Part # BCP1245GExxxx

